

Settling the Complexity of Local Max-Cut (Almost) Completely *

Robert Elsässer and Tobias Tscheuschner

University of Paderborn
Faculty of Computer Science, Electrical Engineering and Mathematics
{elsa, chessy}@upb.de

Abstract. We consider the problem of finding a local optimum for MAX-CUT with FLIP-neighborhood, in which exactly one node changes the partition. Schäffer and Yannakakis (SICOMP, 1991) showed \mathcal{PLS} -completeness of this problem on graphs with unbounded degree. On the other side, Poljak (SICOMP, 1995) showed that in cubic graphs every FLIP local search takes $O(n^2)$ steps, where n is the number of nodes. Due to the huge gap between degree three and unbounded degree, Ackermann, Röglin, and Vöcking (JACM, 2008) asked for the smallest $d \in \mathbb{N}$ for which the local MAX-CUT problem with FLIP-neighborhood on graphs with maximum degree d is \mathcal{PLS} -complete. In this paper, we prove that the computation of a local optimum on graphs with maximum degree five is \mathcal{PLS} -complete. Thus, we solve the problem posed by Ackermann et al. almost completely by showing that d is either four or five (unless $\mathcal{PLS} \subseteq \mathcal{P}$).

On the other side, we also prove that on graphs with degree $O(\log n)$ every FLIP local search has probably polynomial smoothed complexity. Roughly speaking, for any instance, in which the edge weights are perturbed by a (Gaussian) random noise with variance σ^2 , every FLIP local search terminates in time polynomial in n and σ^{-1} , with probability $1 - n^{-\Omega(1)}$. Putting both results together, we may conclude that although local MAX-CUT is likely to be hard on graphs with bounded degree, it can be solved in polynomial time for slightly perturbed instances with high probability.

Key words: Max-Cut, PLS, graphs, local search, smoothed complexity

1 Introduction

For an undirected graph $G = (V, E)$ with weighted edges $w : E \rightarrow \mathbb{N}$ a cut is a partition of V into two sets V_1, V_2 . The weight of the cut is the sum of the weights of the edges connecting nodes between V_1 and V_2 . The MAX-CUT problem asks for a cut of maximum weight. Computing a maximum cut is one of the most famous problems in computer science and is known to be \mathcal{NP} -complete even on graphs with maximum degree three [8]. For a survey of MAX-CUT including applications see [17].

A frequently used approach of dealing with hard combinatorial optimization problems is local search. In local search, to every solution there is assigned a set of neighbor solutions, i. e. a neighborhood. The search begins with an initial solution and iteratively moves to better neighbors until no better neighbor can be found. For a survey of local search, we refer to [14]. To encapsulate many local search problems, Johnson et al. [9] introduced the complexity class \mathcal{PLS} (polynomial local search) and initially showed \mathcal{PLS} -completeness for the CIRCUIT-FLIP problem. Schäffer and Yannakakis [22] showed \mathcal{PLS} -completeness for many popular local search problems including the local MAX-CUT problem with FLIP-neighborhood – albeit their reduction builds graphs with linear degree in the worst case. Moreover, they introduced the notion of so called *tight* \mathcal{PLS} -reductions which preserve not only the existence of instances and initial solutions that are exponentially many improving steps away from any local optimum but also the \mathcal{PSPACE} -completeness of the computation of a local optimum reachable by improving steps from a given solution.

In a recent paper Monien and Tscheuschner [15] showed the two properties that are preserved by tight \mathcal{PLS} -completeness proofs for the local MAX-CUT problem on graphs with maximum degree four. However, their proof did not use a \mathcal{PLS} -reduction; they left open whether the local MAX-CUT problem is \mathcal{PLS} -complete on graphs with maximum degree four. For cubic graphs, Poljak [16] showed that any FLIP-local search takes $O(n^2)$ improving steps, where Loebel [13] earlier showed that a local optimum can be found in polynomial time using an approach different from local search. Thus, it is unlikely that computing a local optimum is \mathcal{PLS} -complete on graphs with maximum degree three.

* Partially supported by the German Research Foundation (DFG) Priority Programme 1307 “Algorithm Engineering”.

Due to the huge gap between degree three and unbounded degree, Ackermann et al. [2] asked for the smallest d such that on graphs with maximum degree d the computation of a local optimum is \mathcal{PLS} -complete. In this paper, we show that d is either four or five (unless $\mathcal{PLS} \subseteq \mathcal{P}$), and thus solve the above problem almost completely. A related problem has been considered by Krentel [10]. He showed \mathcal{PLS} -completeness for a satisfiability problem with trivalent variables, a clause length of at most four, and maximum occurrence of the variables of three.

Our result has impact on many other problems, since the local MAX-CUT has been the basis for many \mathcal{PLS} -reductions in the literature. Some of these reductions directly carry over the property of maximum degree five in some sense and result in \mathcal{PLS} -completeness of the corresponding problem even for very restricted sets of feasible inputs. In particular, \mathcal{PLS} -completeness follows for the MAX-2SAT problem [22] with FLIP-neighborhood, in which exactly one variable changes its value, even if every variable occurs at most ten times. \mathcal{PLS} -completeness also follows for the problem of computing a Nash Equilibrium in CONGESTION GAMES (cf. [6], [2]) in which each strategy contains at most five resources. The problem to PARTITION [22] a graph into two equally sized sets of nodes by minimizing or maximizing the weight of the cut, where the maximum degree is six and the neighborhood consists of all solutions in which two nodes of different partitions are exchanged, is also \mathcal{PLS} -complete. Moreover, our \mathcal{PLS} -completeness proof was already helpful showing a complexity result in hedonic games [7].

In this paper, we also consider the smoothed complexity of any FLIP local search on graphs in which the degrees are bounded by $O(\log n)$. This performance measure has been introduced by Spielman and Teng in their seminal paper on the smoothed analysis of the Simplex algorithm [20]¹. Since then, a large number of papers deal with the smoothed complexity of different algorithms. In most cases, smoothed analysis is used to explain the speed of certain algorithms in practice, which have an unsatisfactory running time according to their worst case complexity.

The smoothed measure of an algorithm on some input instance is its expected performance over random perturbations of that instance, and the smoothed complexity of an algorithm is the maximum smoothed measure over all input instances. In the case of an LP, the goal is to maximize $z^T x$ subject to $Ax \leq b$, for given vectors z , b , and matrix A , where the entries of A are perturbed by Gaussian random variables with mean 0 and variance σ^2 . That is, we add to each entry $a_{i,j}$ some value $\max_{i,j} a_{i,j} \cdot y_{i,j}$, where $y_{i,j}$ is a Gaussian random variable with mean 0 and standard deviation σ . Spielman and Teng showed that an LP, which is perturbed by some random noise as described before, has expected running time polynomial in n , m , and σ . This result has further been improved by Vershynin [23]. The smoothed complexity of other linear programming algorithms has been considered in e.g. [3], and quasi-concave minimization was studied in [11].

Several other algorithms from different areas have been analyzed w. r. t. their smoothed complexity (see [21] for a comprehensive description). Two prominent examples of local search algorithms with polynomial smoothed complexity are 2-opt TSP [5] and k -means [1]. We also mention here the papers of Beier, Röglin, and Vöcking [4,19] on the smoothed analysis of integer linear programming. They showed that if Π is a certain class of integer linear programs, then Π has an algorithm of probably polynomial smoothed complexity² iff $\Pi_u \in ZPP$, where Π_u is the unary representation of Π , and ZPP denotes the class of decision problems solvable by a randomized algorithm with polynomial expected running time that always returns the correct answer. The results of [4,19] imply that e.g. 0/1-knapsack, constrained shortest path, and constrained minimum weighted matching have probably polynomial smoothed complexity. Unfortunately, the results of these papers cannot be used to settle the smoothed complexity of local MAX-CUT.

Overview In section 3, we introduce a technique by which we substitute graphs whose nodes of degree greater than five have a certain type – we will call these nodes **comparing** – by graphs of maximum degree five. In particular, we show that certain local optima in the former graphs induce unique local optima in the latter ones. In section 4 we show an overview of the proof of the \mathcal{PLS} -completeness of computing a local optimum of MAX-CUT on graphs with maximum degree five by reducing from the \mathcal{PLS} -complete problem CIRCUITFLIP. In a nutshell, we map instances of CIRCUITFLIP to graphs whose nodes of degree greater than five are comparing. Some parts of the graphs are adjustments of subgraphs of the \mathcal{PLS} -completeness proof

¹ For this work, Spielman and Teng was awarded the Gödel Prize in 2008.

² For the definition of probably polynomial smoothed complexity see Section 5.

of [22]. Then, using our technique, we show that local optima for these graphs induce local optima in the corresponding instances of CIRCUITFLIP.

In section 5 we show that on graphs with degree $O(\log n)$ local MAX-CUT has probably polynomial smoothed complexity. To obtain this result, we basically prove that every improving step w. r. t. the FLIP-neighborhood increases the cut by at least a polynomial value in n and/or σ , with high probability.

2 Preliminaries

A graph G together with a 2-partition P of V is denoted by G_P . We let $c_{G_P} : V \rightarrow \{0, 1\}$ with $c_{G_P}(u) = 1$ if and only if $u \in V_1$ in G_P . We let $c_{G_P}(u)$ be the **color** of u in G_P , where u is white if $c_{G_P}(u) = 0$ and black otherwise. If the considered graph is clear from the context then we also just write $c_P(v)$ and if even the partition is clear then we omit the whole subscript. For convenience we treat the colors of the nodes also as truth values, i. e. black corresponds to *true* and white to *false*. For a vector v of nodes we let $c(v)$ be the vector of colors induced by c . We say that an edge $\{u, v\}$ is **in the cut** in P if $c_P(u) \neq c_P(v)$. For a node u we say that u **flips** if it changes the partition. A node u is **happy** in G_P if a flip of u does not increase the weight of the cut, and **unhappy** otherwise. Since we consider weighted graphs, we also say that a flip increases the cut if it increases the weight of the cut. A partition P is a **local optimum** if all nodes in G_P are happy.

A local search problem Π consists of a set of instances \mathcal{I} , a set of feasible solutions $\mathcal{F}(I)$ for every instance $I \in \mathcal{I}$, and an objective function $f : \mathcal{F}(I) \rightarrow \mathbb{Z}$. In addition, every solution $s \in \mathcal{F}(I)$ has a neighborhood $\mathcal{N}(s, I) \subseteq \mathcal{F}(I)$. For an instance $I \in \mathcal{I}$, the problem is to find a solution $s \in \mathcal{F}(I)$ such that for all $s' \in \mathcal{N}(s, I)$ solution s' does not have a greater value than s with respect to f in case of maximization and not a lower value in case of minimization.

A local search problem Π is in the class **PLS** [9] if the following three polynomial time algorithms exist: algorithm A computes for every instance $I \in \mathcal{I}$ a feasible solution $s \in \mathcal{F}(I)$, algorithm B computes for every $I \in \mathcal{I}$ and $s \in \mathcal{F}(I)$ the value $f(s)$, and algorithm C returns for every $I \in \mathcal{I}$ and $s \in \mathcal{F}(I)$ a better neighbor solution $s' \in \mathcal{N}(s, I)$ if there is one and “locally optimal” otherwise. A problem $\Pi \in \mathcal{PLS}$ is **PLS-reducible** to a problem $\Pi' \in \mathcal{PLS}$ if there are the following polynomial time computable functions Φ and Ψ . The function Φ maps instances I of Π to instances of Π' and Ψ maps pairs (s, I) , where s is a solution of $\Phi(I)$, to solutions of I , such that for all instances I of Π and local optima s^* of $\Phi(I)$ the solution $\Psi(s^*, I)$ is a local optimum of I . Finally, a problem $\Pi \in \mathcal{PLS}$ is **PLS-complete** if every problem in \mathcal{PLS} is PLS-reducible to Π .

In our technique, as well as in the PLS-completeness proof, we make use of a result of Monien and Tscheuschner [15]. They showed a property for a set of graphs containing two certain types of nodes of degree four. Since we do not need their types in this paper, we omit the restrictions on the nodes and use the following weaker proposition.

Lemma 1 ([15]). *Let C_f be a boolean circuit with N gates which computes a function $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$. Then, using $O(\log N)$ space, one can compute a graph $G_f = (V_f, E_f)$ with maximum degree four containing nodes $s_1, \dots, s_n, t_1, \dots, t_m \in V_f$ of degree one such that for the vectors $s := (s_1, \dots, s_n), t := (t_1, \dots, t_m)$ we have $f(c_P(s)) = c_P(t)$ in every local optimum P of G_f .*

Definition 1. *For a polynomial time computable function f we say that $G_f = (V_f, E_f)$ as constructed in Lemma 1 is the graph that **looks** at the input nodes $s_i \in V_f$ and **biases** the output nodes $t_i \in V_f$ to take the colors induced by f .*

Usage of Lemma 1 Notice first that G_f can be constructed in logarithmic space and thus polynomial time for any polynomial time computable function f . In the rest of the paper we use the graph G_f for several functions f and we will scale the weights of its edges. Then, the edges of G_f give incentives of appropriate weight to certain nodes of those graphs to which we add G_f . The incentives bias the nodes to take the colors induced by f . We already point out that for any node v we will introduce at most one subgraph that biases v . Moreover, the unique edge $e = \{u, v\}$ incident to a biased node v that is an edge of the subgraph that biases v will in many cases have the lowest weight among the edges incident to v . In particular, the weight of e will then be chosen small enough such that the color of v , in local optima, depends on the color of u if and only if v is indifferent with respect to the colors of the other nodes adjacent to v . Note that in local optima the node u has the opposite color as the color to which v is biased according to f .

3 Substituting certain nodes of unbounded degree

Definition 2. Let $G = (V, E)$ be a graph. A node $v \in V$ is called **comparing** if there is an $m \in \mathbb{N}$ such that

- (i) v is adjacent to exactly $2m+1$ nodes $u_1^1, u_1^2, u_2^1, u_2^2, \dots, u_m^1, u_m^2, u \in V \setminus \{v\}$ with edge weights a_1, \dots, a_m, δ , as shown in Figure 1,
- (ii) u is a node of a subgraph $G' = (V', E')$ of G that looks at a subset of $V \setminus \{u, v\}$ and biases v ,
- (iii) $a_i \geq 2a_{i+1}$ for all $1 \leq i < m$ and $a_m \geq 2\delta$.

The subgraph G' is called the **biasser** of v . For u_i^j with $1 \leq i \leq m$, $1 \leq j \leq 2$ we call the node u_i^k with $1 \leq k \leq 2$ and $k \neq j$ adjacent to v via the unique edge with the same weight as $\{u_i^j, v\}$ the **counterpart** of u_i^j with respect to v .

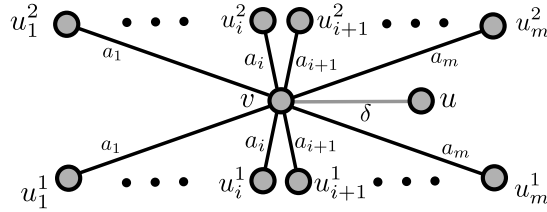


Fig. 1. Node v is a comparing node.

The name of the comparing node stems from its behaviour in local optima. If we treat the colors of the neighbors u_1^1, \dots, u_m^1 of v as a binary number a , with u_1^1 being the most significant bit, and the colors of u_1^2, \dots, u_m^2 as the bitwise complement of a binary number b then, in a local optimum, the comparing node v is white if $a > b$, it is black if $a < b$, and if $a = b$ then v has the color to which it is biased by its biasser. In this way, the color of v “compares” a and b in local optima.

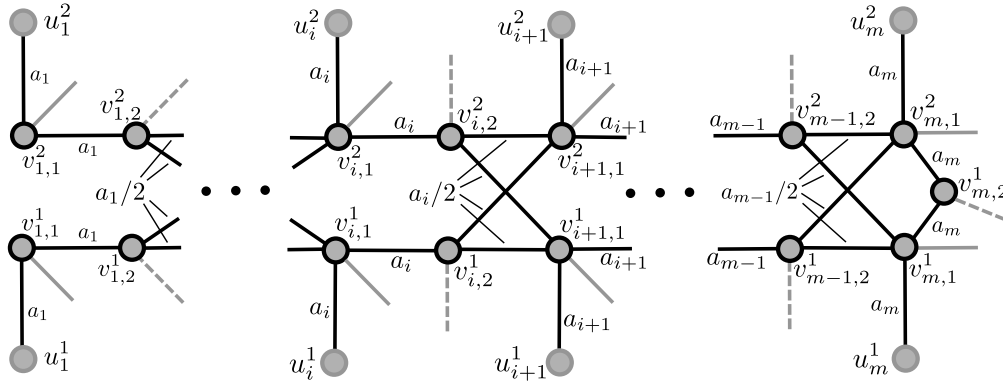


Fig. 2. The gadget that substitutes a comparing node v .

In the following, we let $G = (V, E)$ be a graph and $v \in V$ be a comparing node with adjacent nodes and incident edges as in Figure 1. We say that we **degrade** v if we remove v and its incident edges and add the following nodes and edges. We introduce nodes $v_{i,j}^k$ for $1 \leq i < m$, $1 \leq j \leq 2$, $1 \leq k \leq 2$, nodes $v_{m,1}^k$ for $1 \leq k \leq 2$, and $v_{i,2}^1$ with edges and weights as depicted in Figure 2 – the nodes u_i^j in Figure 2 have gray circumcircles to indicate that they, in contrast to the other nodes, also occur in G . Furthermore, we add a subgraph G'' that looks at u and biases all nodes $v_{i,1}^k$ to the opposite of the color of u (this is illustrated by short gray edges in Figure 2) and the nodes $v_{i,2}^k$ to the color of u (short gray dashed edges). The weights of

the edges of G'' are scaled such that each of them is strictly smaller than δ . Note that due to the scaling the color of the unique node of G'' adjacent to u does not affect the happiness of u – node u is therefore not depicted in Figure 2 anymore. We let $\mathbf{G}(\mathbf{G}, \mathbf{v})$ be the graph obtained from G by degrading v and we call v **weakly indifferent** in a partition P if $c_P(u_i^1) \neq c_P(u_i^2)$ for all $1 \leq i \leq m$. If v is not weakly indifferent then we call the two nodes u_i^1, u_i^2 adjacent to v via the edges with highest weight for which $c_P(u_i^1) = c_P(u_i^2)$ the **decisive neighbors** of v in P . We let $V_{com} \subseteq V$ be the set of comparing nodes of V , and for a partition P of the nodes of $G(G, v)$ we let $\mathbf{col}_P : V_{com} \rightarrow \{0, 1\}$ be the partial function defined by

$$\mathbf{col}_P(v) = \begin{cases} 0, & \text{if for all } i, j : c_P(v_{i,1}^j) = 0 \text{ and } c_P(v_{i,2}^j) = 1, \\ 1, & \text{if for all } i, j : c_P(v_{i,1}^j) = 1 \text{ and } c_P(v_{i,2}^j) = 0. \end{cases}$$

We say that a comparing node v has the color $\kappa \in \{0, 1\}$ in a partition P if $\mathbf{col}_P(v) = \kappa$.

Theorem 1. *Let $G = (V, E)$ be a graph, $v \in V$ a comparing node, its adjacent nodes and incident edges as in Figure 1, P be a local optimum of G such that in P the biasser of v biases v to $c_P(v)$, i. e. $c_P(u) \neq c_P(v)$. Let P' be a partition of the nodes of $G(G, v)$ such that $c_P(w) = c_{P'}(w)$ for all $w \in V \setminus \{v\}$. Then, P' is a local optimum if and only if $c_P(v) = \mathbf{col}_{P'}(v)$.*

Note the restriction that in the local optimum P the biasser of v biases v to the color that v in fact has in P and *not* to the opposite. In the \mathcal{PLS} -completeness proof in section 4 the biasser of any comparing node v is designed to bias v to the color that v has in a local optimum P due the colors of its neighbors. Then, we can use Theorem 1 to argue about $\mathbf{col}_P(v)$ in $G(G, v)$.

Proof. Let $\kappa \in \{0, 1\}$ be the color to which v is biased by its biasser in P , i. e. $\kappa := c_P(v)$. For all i, j we call the color of $v_{i,1}^j$ **correct** if $c_{P'}(v_{i,1}^j) = \kappa$ and we call the color of $v_{i,2}^j$ correct if $c_{P'}(v_{i,2}^j) = \bar{\kappa}$. Moreover, we call $v_{i,j}^k$ correct for any i, j, k if it has its correct color.

“ \Rightarrow ”: Let P' be a local optimum. Note that each node $v_{i,j}^k$ is biased by an edge with weight lower than δ to its correct color. Therefore, to show that it is correct in the local optimum P' , it suffices to show that it gains at least half of the sum of weights of the incident edges with weight greater than δ if it is correct. We prove the Theorem by means of the following Lemmas which are each proven via straightforward inductive arguments.

Lemma 2. *Let $q \leq m$ and $c_P(u_i^1) = \bar{\kappa}$ for all $i \leq q$. Then, $v_{i,1}^1$ and $v_{i,2}^1$ are correct for all $i \leq q$.*

Proof. We prove the claim by induction on i . Due to $c_P(u_1^1) = \bar{\kappa}$ we get the correctness of $v_{1,1}^1$. For each $i \leq q$ the correctness of $v_{i,1}^1$ implies the correctness of $v_{i,2}^1$. Moreover, for each $i < q$ the correctness of $v_{i,2}^1$ together with $c_P(u_{i+1}^1) = \bar{\kappa}$ implies the correctness of $v_{i+1,1}^1$. \square

Lemma 3. *Let $q \leq m$, node $v_{i,1}^1$ and $v_{i,2}^1$ be correct for all $i \leq q$, and $v_{q,1}^2$ be correct. Then, $v_{i,1}^2$ and $v_{i,2}^2$ are correct for all $i < q$.*

Proof. We prove the claim by induction on i . Node $v_{q,1}^2$ and $v_{q,1}^1$ are correct by assumption. For each $i < q$ node $v_{i,2}^2$ is correct if $v_{i+1,1}^2$ is correct since $v_{i+1,1}^1$ is correct by assumption. Moreover, for each $1 < i < q$ node $v_{i,1}^2$ is correct if $v_{i,2}^2$ is correct since $v_{i-1,2}^1$ is correct by assumption. Finally, node $v_{1,1}^2$ is correct if $v_{1,2}^2$ is correct. \square

Lemma 4. *Let $q \leq m$. If $v_{q,1}^1$ and $v_{q,1}^2$ are correct then $v_{i,j}^k$ is correct for any j, k , and $q \leq i \leq m$.*

Proof. If $q = m$ then the correctness of $v_{m,1}^1$ implies the correctness of $v_{m,2}^1$. The case $q < m$ is done by induction on i . Node $v_{q,1}^1$ and $v_{q,1}^2$ are correct by assumption. Assume that $v_{i,1}^1$ and $v_{i,1}^2$ are correct for an arbitrary $q \leq i < m$. Then, the nodes $v_{i,2}^1$ and $v_{i,2}^2$ are correct whereafter the correctness of $v_{i+1,1}^1$ and $v_{i+1,1}^2$ follows. Finally, the correctness of $v_{m,1}^1$ implies the correctness of $v_{m,2}^1$. \square

We first consider the case that v is weakly indifferent. Then, for each i at least one of the nodes u_i^1 and u_i^2 has the color κ . Due to the symmetry between the nodes $v_{i,j}^1$ and $v_{i,j}^2$ we may assume w. l. o. g. that $c_P(u_i^1) = \bar{\kappa}$ for all i . Then, Lemma 2 implies that $v_{i,1}^1$ and $v_{i,2}^1$ are correct for all i . Then, the correctness of

$v_{m,2}^1$ and $v_{m-1,2}^1$ together imply the correctness of $v_{m,1}^2$. Then, Lemma 3 implies the correctness of $v_{i,1}^2$ and $v_{i,2}^2$ for all $i < m$.

Now assume that v is not weakly indifferent and let u_q^1 and u_q^2 be the decisive neighbors of v . As in the previous case we assume w. l. o. g. that $c_P(u_i^1) = \bar{\kappa}$ for all $i \leq q$. Then, due to Lemma 2 node $v_{i,1}^1$ and $v_{i,2}^1$ are correct for all $i \leq q$. If $q = 1$ then $c(u_1^2) = \bar{\kappa}$ implies the correctness of $v_{1,1}^2$ – recall that by assumption v is biased to the opposite color of the color of the decisive nodes. On the other hand, if $q > 1$ then the correctness of $v_{q-1,2}^1$ and $c(u_q^2) = \bar{\kappa}$ together imply the correctness of $v_{q,1}^2$. Then, Lemma 3 implies the correctness of $v_{i,1}^2$ and $v_{i,2}^2$ for all $i < q$. Finally, Lemma 4 implies the correctness of $v_{i,j}^k$ for all j, k , and $q \leq i \leq m$.

“ \Leftarrow ”: Assume, that every node $v_{i,j}^k$ is correct. As we have seen in “ \Rightarrow ” $v_{i,j}^k$ is happy then. Moreover, each u_i^j is also happy since its neighbors have the same colors as in the local optimum P – recall that if $v_{i,1}^j$ is correct it has the same color in P' as v in P . The colors of the remaining nodes are unchanged. Therefore, P' is a local optimum. This finishes the proof of Theorem 1. \square

4 Proof of \mathcal{PLS} -Completeness

Our reduction bases on the following \mathcal{PLS} -complete problem CIRCUITFLIP (in [9] it is called FLIP, which we avoid in this paper since the neighborhood of MAX-CUT has the same name).

Definition 3 ([9]). *An instance of CIRCUITFLIP is a boolean circuit C with n input bits and m output bits. A feasible solution of CIRCUITFLIP is a vector $v \in \{0, 1\}^n$ of input bits for C and the value of a solution is the output of C treated as a binary number. Two solutions are neighbors if they differ in exactly one bit. The objective is to maximize the output of C .*

Theorem 2. *The problem of computing a local optimum of the MAX-CUT problem on graphs with maximum degree five is \mathcal{PLS} -complete.*

Proof. We reduce from the \mathcal{PLS} -complete problem CIRCUITFLIP. Let C be an instance of CIRCUITFLIP with input variables X_1, \dots, X_n , outputs C_1, \dots, C_m , and gates G_N, \dots, G_1 . W. l. o. g. we make the following assumptions. Each input variable occurs exactly once in exactly one gate. All gates are NOR-gates with a fanin of 2 and are topologically sorted such that $i > j$ if G_i is an input of G_j . For the sake of simplicity, we denote G_i also as the output of gate G_i . The two inputs of a gate G_i are denoted by $I_1(G_i)$ and $I_2(G_i)$, i. e. a gate G_i computes correctly if and only if $G_i = \neg(I_1(G_i) \vee I_2(G_i))$. For no gate G_i we have $I_1(G_i) = I_2(G_i)$. The gates G_1, \dots, G_m are the output of C where G_m is the most significant bit and G_{m+1}, \dots, G_{2m} compute the corresponding negations of the output bits. The gates $G_{2m+1}, \dots, G_{2m+n}$ and $G_{2m+n+1}, \dots, G_{2m+2n}$ return the same better neighbor solution if there is one and return X_1, \dots, X_n otherwise. Finally, let $C(x)$ be the output of C on input $x \in \{0, 1\}^n$ and $w(x)$ be the better neighbor of x computed by C on input x and assume w. l. o. g. $N > 20$ and $m \geq n > 4$.

The proof in a nutshell: From C we construct a graph G_C consisting of two isomorphic subgraphs G_C^0, G_C^1 representing copies of C – the overall structure of our proof is inspired by [10]. For each gate G_i in C there is a subgraph S_i^κ for $\kappa \in \{0, 1\}$ in G_C . The subgraphs S_i^κ are taken from [22] and adjusted such that they have maximum degree five without changing local optima. In particular, each S_i^κ contains a comparing node g_i^κ whose color represents the output of G_i . To maintain a maximum degree of five we assume that g_i^κ is degraded in G_C and argue via Theorem 1 about its color in local optima. Then, the colors of the nodes of S_i^κ , in local optima, either behave as a NOR-gate or have a **reset** state, i. e. a state in which each input node of S_i^κ is indifferent w. r. t. its neighbors in S_i^κ . For each $\kappa \in \{0, 1\}$ we have a subgraph T^κ that looks at g_i^κ for $2m+1 \leq i \leq 2m+n$, i. e. at the improving solution, and biases each input node of G_C^κ to the color of its corresponding g_i^κ . Finally, we have a subgraph that looks at the input nodes of G_C^0, G_C^1 , decides whose input results in a greater output w. r. t. C – this subgraph is called **winner** as opposed to the **loser** which is the other subgraph – and biases the subgraphs S_i^κ of the winner to behave like NOR-gates and the subgraphs of the loser to take the reset state. Then, we show that the colors of the subgraphs S_i^κ of the winner in fact reflect the correct outputs w. r. t. their inputs and that the input nodes of the loser in fact are indifferent w. r. t. their neighbors in the subgraphs S_i^κ . Then, due to the bias of T^κ , the input nodes of the loser take the colors of the improving neighbor computed by the winner whereafter the loser becomes the

new winner. Hence, the improving solutions switch back and forth between the two copies until the colors of the input nodes of both copies are local optima and the copies return their input as improving solution. Then, the colors of the input nodes induce a local optimum of C .

Before turning into the details we introduce some notations w. r. t. G_C . We let x_i^κ be the input nodes of G_C^κ , $w_{i,1}^\kappa := g_{2m+i}^\kappa$, $w_{i,2}^\kappa := g_{2m+n+i}^\kappa$ for $1 \leq i \leq n$, and $\hat{g}_i^\kappa := g_{m+i}^\kappa$ for $1 \leq i \leq m$. Each subgraph G_C^κ also contains nodes y_i^κ , z_i^κ for $0 \leq i \leq 2N+1$ and λ_i^κ for $1 \leq i \leq n$ which induce vectors y^κ , z^κ , and λ^κ . Moreover, we let x^κ be the vector of nodes induced by x_i^κ for $1 \leq i \leq n$.

We will introduce the nodes and edges of G_C via so called components. A component of G_C is a tuple (V'_C, E'_C) with $V'_C \subseteq V_C$ and $E'_C \subseteq E_C$. The components of G_C have fourteen types: type 1 up to type 14, where we say that the nodes, edges, and weights of the edges of the components have the same types as their corresponding components. We will explicitly state weights for the edges of type 2 up to 7. However, the weights of these components are only stated to indicate the relations between edge weights of the same type. The only edge weights that interleave between two different types are those of type 3 and 4. The edges of type 3 and 4 are scaled by the same number. For all other types we assume that their weights are scaled such that the weight of an edge of a given type is greater than four times the sum of the weights of the edges of higher types combined. Note that for these types a lower type implies a higher edge weight. To distinguish between the meaning of the explicitly stated edge weights and the final edge weights, i. e. the weights resulting by the scale, we will speak of the explicitly stated weights of **relative edge weights**.

The components of some types are introduced via drawings. In the drawings, the thick black edges and the nodes with black circumspheres are nodes counted among the components of the introduced type. Gray edges and nodes with gray circumspheres are of a different type than the component introduced in the corresponding drawing and are only (re-)drawn to simplify the verification of the proofs for the reader – in particular the condition that each node is of maximum degree five. If for a gray edge there is no explicit relative weight given then the edge is among the types 8 – 14. If a gray edge is dotted then it is of higher type than the non-dotted gray edges of the same drawing. If a node has a black or a white filling then it is of type 1. These nodes are also (re-)drawn in components of type higher than 1.

Type 1 is to provide the constants 0 and 1 for the components of higher type. It contains nodes s, t which are connected by an edge with a weight that is greater than the sum of all other edges in E_C . Assume w. l. o. g. $c(s) = 0$ and let S and T be the sets of nodes representing the constants 0 and 1. Type 1 looks at s and biases the nodes of S to the color of s and the nodes of T to the opposite. In the following we assume for each constant introduced in components of higher types there is a separate node in the sets S, T .

Type 2 contains the nodes d^0, d^1, u^0, u^1 – we will see later that d^0 and d^1 are comparing nodes – with edges and relative weights as depicted in Figure 3. The purpose of these edges is – together with the edges of type 9 and 10 – to guarantee that d^0 and d^1 are not both black in local optima. The nodes d^0 and d^1 are adjacent to many nodes of higher type, and have a degree greater than five.

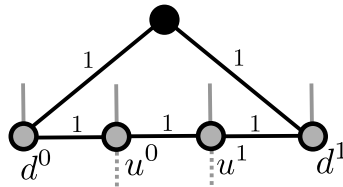


Fig. 3. The component of type 2

The components of type 3 to 7 are to represent the two subgraphs G_C^0 and G_C^1 . The components are very similar to certain clauses of [22]. There are three differences between our components and their clauses. First, we omit some nodes and edges to obtain a maximum degree of five for all nodes different from g_i^κ , $I_1(g_i^\kappa)$, and $I_2(g_i^\kappa)$. Second, we use different edge weights. However, the weights are manipulated in a way such that the happiness of each node for given colors of the corresponding adjacent nodes is the same as in [22]. Third, we add nodes that we bias and to which we look at. Their purpose is to derive the color that a comparing node g_i^κ would have if it was a single node. This color is used to bias g_i^κ such that Theorem 1 implies either $col(g_i^\kappa) = 0$ or $col(g_i^\kappa) = 1$.

Type 3 consists of subgraphs S_i^κ which are to represent the gates G_i of C . For gates whose inputs are not inputs of G_C^κ , they are depicted in Figure 4. Together with d^0 and d^1 , the nodes g_i^κ (and $I_k(g_j^\kappa)$ respectively) are the only nodes which have a degree greater than five – we will see later that they are also comparing. For each gate G_i^κ whose inputs are inputs of G_C^κ we take the same components as for those gates whose inputs are not inputs of G_C^κ but make the following adjustment. We omit the edges $\{I_1(g_i^\kappa), 0\}$ and $\{I_2(g_i^\kappa), 1\}$ and subtract their relative weights from the edges $\{I_1(g_i^\kappa), 1\}$ and $\{I_2(g_i^\kappa), 0\}$ respectively, i. e. their relative weights are $2^{10i+7} - 2^{10i-5}$ and $2^{10i-5} - 2^{10i-1}$. Note that the adjustment does not change the happiness of the nodes $I_1(g_i^\kappa)$ and $I_2(g_i^\kappa)$ for any given colors of themselves and their neighbors. We call the edges $\{g_i^\kappa, u_{i,j}^\kappa\}$ for $j \in \{2, 3, 6, 7, 10, 11\}$ **corresponding** to g_i^κ .

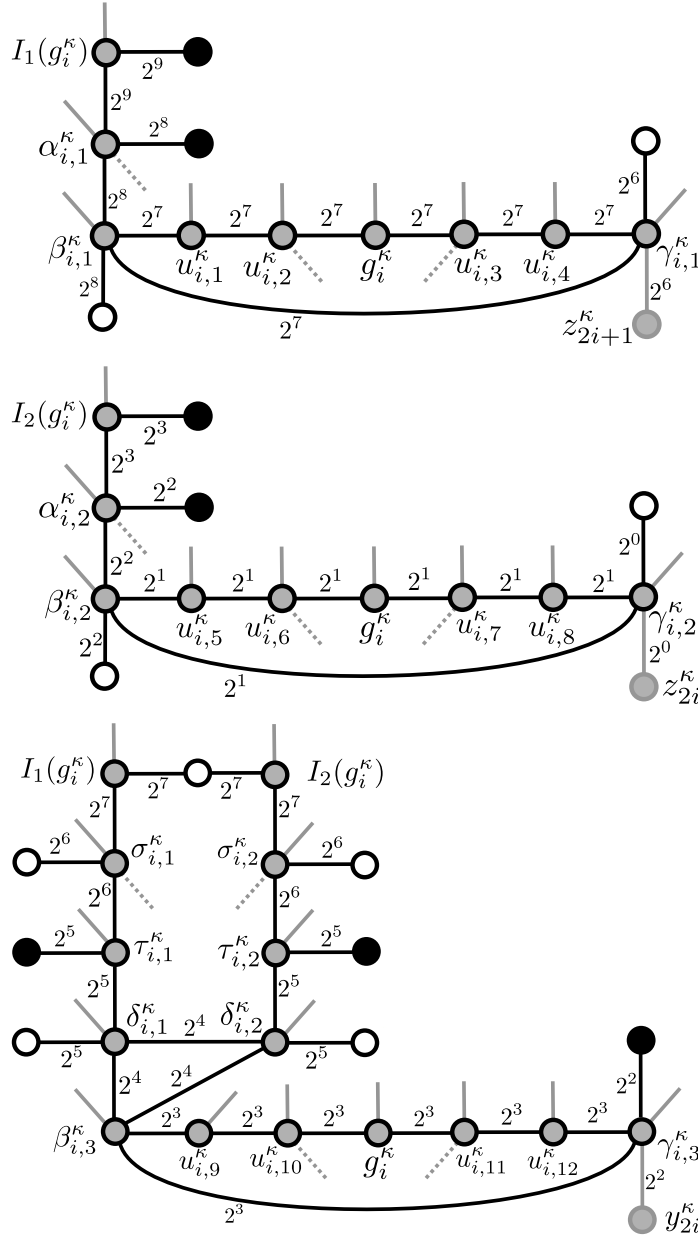


Fig. 4. The components of type 3; extra factor for relative edge weights: 2^{10i-2}

Type 4 (Figure 5) checks whether the outputs of the gates represented by the components of type 3 are correct and gives incentives to nodes of other components depending on the result. As in [22] we say that the **natural value** of the nodes y_i^κ is 1 and the natural value of the nodes z_i^κ is 0. The nodes $y_{N+1}^\kappa, z_{N+1}^\kappa, \dots, y_2^\kappa, z_2^\kappa$ check the correct computation of the corresponding gates and give incentives to their corresponding gates depending on whether the previous gates are correct. The nodes $y_1^\kappa, z_1^\kappa, y_0^\kappa, z_0^\kappa$ are to give incentives to d^0, d^1 depending on whether all gates are correct. Recall that the weights of the edges of type 4 are the only weights that interleave with weights of edges of a higher type, namely with those of type 3.

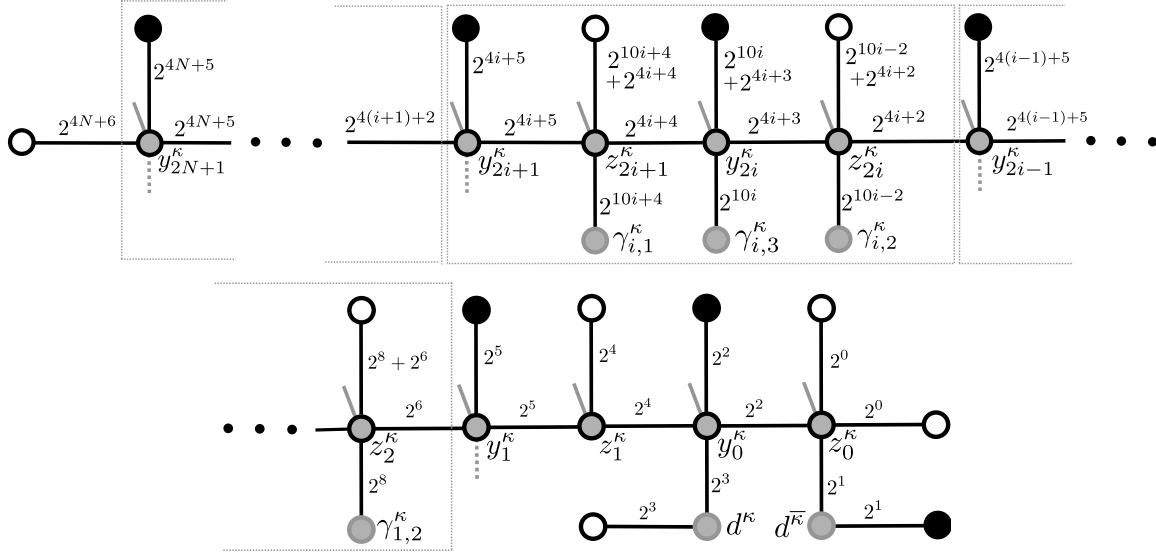


Fig. 5. The components of type 4

Type 5 contains the nodes and edges as depicted in Figure 6 for $1 \leq i \leq m$ and edges $\{1, d^0\}$, $\{1, d^0\}$, $\{0, d^1\}$, and $\{0, d^1\}$ of relative weight 1 – these edges are not depicted. The aim of the component is twofold. On the one hand it is to incite that one of the nodes d^0 and d^1 to become black for which the output of the corresponding copy G_C^0 and G_C^1 is smaller and the other one to become white. On the other hand, the edges $\{1, d^0\}$, $\{1, d^0\}$, $\{0, d^1\}$, and $\{0, d^1\}$ are to break the tie in favor of G_C^0 if the outputs of G_C^0 and G_C^1 are equal.

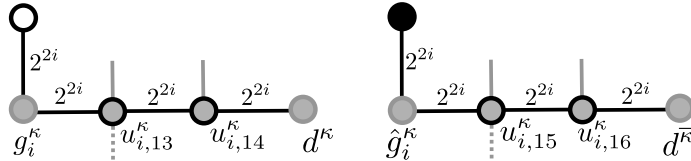


Fig. 6. One part of the component of type 5

Type 6 contains nodes \hat{d}_i^κ for all $1 \leq i \leq n$ with incident edges $\{\hat{d}_i^\kappa, d^\kappa\}$ of relative weight 2^{2i} . These edges are to ensure that $col(d^\kappa) \neq c(\hat{d}_i^\kappa)$ for all i . The component also contains n edges $\{1, d^\kappa\}$ with relative weights 2^{2i} for all $1 \leq i \leq n$ – recall that each constant is represented by a separate node of type 1. These edges are needed for d^κ to be a comparing node.

Type 7 (Figure 7) is to incite the input nodes of $G_C^{\bar{\kappa}}$ to take the color corresponding to the better neighbor computed by $G_C^{\bar{\kappa}}$ if $col(d^\kappa) = 0$. As we will see in Lemma 6 the node $\lambda_i^{\bar{\kappa}}$ has the same color as $w_{i,1}^\kappa$ if $col(w_{i,1}^\kappa) = col(w_{i,2}^\kappa)$ and $col(d^\kappa) = 0$. Moreover, we will see in the same Lemma that $\lambda_i^{\bar{\kappa}}$ has the opposite

color as μ_i^κ in any local optima. Therefore the nodes λ_i^κ and μ_i^κ together with their incident edges, in the case that $w_{i,1}^\kappa = w_{i,2}^\kappa$ and $\text{col}(d^\kappa) = 0$, have the functionality of a subgraph T^κ that looks at the nodes $w_{i,1}^\kappa$ and biases the input nodes of G_C^κ to take the color of their corresponding $w_{i,1}^\kappa$. Concerning the maximum degree of five recall that the number of edges of type 3 incident to x^κ was three due to the adjustment. One edge of type 7 is incident to x^κ and one edge of higher type – depicted as a gray edge in Figure 7 – is incident to x^κ . Thus, x^κ has a degree of five.

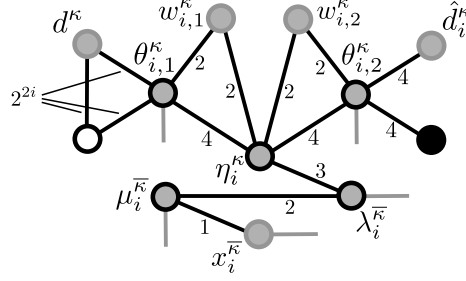


Fig. 7. The components of type 7

The components of type 8 to 14 are subgraphs that look at certain nodes and bias other nodes. No node to which any component looks at is a comparing node. Therefore, all of them must be of degree at most five in our construction. But to some of these nodes more than one component looks at. To maintain a maximum degree of five for these nodes, we assume that the component of the lowest type which looks at such a node v not only biases the nodes of which we state that it biases them but also biases extra nodes v'_1, \dots, v'_k , for $k \in \mathbb{N}$ great enough, to have the same color as v and the components of higher types look at v'_1, \dots, v'_k instead of the original nodes.

Type 8 looks at the vectors x^0, x^1 of nodes representing the inputs of G_0^C and G_1^C and at the vectors λ^0, λ^1 of nodes of type 7 and biases the vectors y^0, z^0, y^1 , and z^1 in the following way. The nodes y_i^0, z_i^0 for all $0 \leq i \leq 2N+1$ are biased to their unnatural value, as defined in type 4, if $C(x^0) < C(x^1)$, $w(x^1) \neq c(x^0)$, and $w(x^1) = c(\lambda^0)$ and to their natural value otherwise. Similarly, y_i^1, z_i^1 are biased to their unnatural value if $C(x^0) \geq C(x^1)$, $w(x^0) \neq c(x^1)$, and $w(x^0) = c(\lambda^1)$ and to their natural value otherwise. The comparison between $C(x^0)$ and $C(x^1)$ is used to decide which circuit is the winner and which one is the loser and the consideration of the other colors is to avoid certain troubling local optima.

The idea behind the next two components is as follows. In any local optimum, we want for the nodes d^0 and d^1 at most one to be black. The immediate idea to reach this would be to use a simple edge between them in the component of type 2 (see Figure 3) without the intermediate nodes u^0 and u^1 . To show – later in the proof – that a comparing node d^κ has a certain color, we want to apply Theorem 1. For this, we need to know the colors of the neighbors adjacent to d^κ via the edges of the highest weight, which includes the color of d^κ . But argue about the color of d^κ via Theorem 1 analogously needs the information about the color of d^κ . To solve this problem, we introduce the intermediate nodes u^0 and u^1 , bias them appropriately and use their colors to bias d^0 and d^1 .

Type 9 looks at y_1^0, y_1^1 , and at the vectors x^0 and x^1 and biases u^0 and u^1 as follows. If $C(x^0) \geq C(x^1)$ then it biases u^0 to the color of y_1^0 and u^1 to the opposite. Otherwise it biases u^1 to the color of y_1^1 and u^0 to the opposite.

Type 10 looks at u^0, u^1, y_1^0, y_1^1 , and at the vectors x^0 and x^1 and biases d^0 and d^1 as follows. If $c(y_1^0) = c(y_1^1) = 0$ then d^0 is biased to the color of u^1 and d^1 to the color of u^0 . If $c(y_1^0) \neq c(y_1^1)$ then d^0 is biased to the color of y_1^1 and d^1 to the opposite. If $c(y_1^0) = c(y_1^1) = 1$ then we distinguish two cases. If $C(x^0) \geq C(x^1)$ then d^0 is biased to 0 and d^1 to 1, otherwise d^0 to 1 and d^1 to 0.

Type 11 is to bias the nodes of type 3 to certain preferred colors depending on whether y_{2i+1}^κ has its natural value. If it has its natural value then it biases the subgraph S_i^κ to colors which reflect the behavior of a NOR-gate for S_i^κ and otherwise it biases them such that the input nodes $I_1(g_i^\kappa)$ and $I_2(g_i^\kappa)$ are indifferent with respect to their neighbors in S_i^κ , i. e. the nodes of S_i^κ are biased to their reset state. In particular the

component looks at y_{2i+1}^κ for $1 \leq i \leq N$ and biases $\alpha_{i,1}^\kappa, \alpha_{i,2}^\kappa, \gamma_{i,1}^\kappa, \gamma_{i,2}^\kappa, \beta_{i,3}^\kappa, \tau_{i,1}^\kappa$, and $\tau_{i,2}^\kappa$ to the color of y_{2i+1}^κ and $\beta_{i,1}^\kappa, \beta_{i,2}^\kappa, \gamma_{i,3}^\kappa, \sigma_{i,1}^\kappa, \sigma_{i,2}^\kappa, \delta_{i,1}^\kappa$, and $\delta_{i,2}^\kappa$ to the opposite.

The aim of the next two components is as follows. We want to bias the comparing nodes g_i^κ such that we can apply Theorem 1 to obtain either $\text{col}(g_i^\kappa) = 1$ or $\text{col}(g_i^\kappa) = 0$. To reach this, we need to know the colors of the nodes adjacent to g_i^κ . For this purpose we introduce – similarly as in the component of type 2 – extra nodes $u_{i,j}^\kappa$, bias them appropriately and use their colors instead.

Type 12 looks at $y_{2i+1}^\kappa, y_{2i-1}^\kappa, \alpha_{i,1}^\kappa$, and $\alpha_{i,2}^\kappa$ and biases $u_{i,1}^\kappa, u_{i,3}^\kappa, u_{i,5}^\kappa, u_{i,7}^\kappa, u_{i,10}^\kappa, u_{i,12}^\kappa$ to white and $u_{i,2}^\kappa, u_{i,4}^\kappa, u_{i,6}^\kappa, u_{i,8}^\kappa, u_{i,9}^\kappa, u_{i,11}^\kappa$ to black if $c(y_{2i+1}^\kappa) = c(y_{2i-1}^\kappa)$. Otherwise, $u_{i,3}^\kappa, u_{i,4}^\kappa, u_{i,7}^\kappa, u_{i,8}^\kappa, u_{i,11}^\kappa, u_{i,12}^\kappa$ are biased to their respective opposite and the biases of the remaining nodes split into the following cases. Node $u_{i,1}^\kappa$ is biased to $c(\alpha_{i,1}^\kappa)$ and $u_{i,2}^\kappa$ to the opposite. Similarly, $u_{i,5}^\kappa$ is biased to $c(\alpha_{i,2}^\kappa)$ and $u_{i,6}^\kappa$ to the opposite. Finally, $u_{i,9}^\kappa$ is biased to $c(\alpha_{i,2}^\kappa) \wedge c(\alpha_{i,1}^\kappa)$ and $u_{i,10}^\kappa$ to the opposite.

Type 13 looks for all $1 \leq i \leq m$ at $y_{2i-1}^\kappa, \alpha_{i,1}^\kappa$, and $\alpha_{i,2}^\kappa$ and biases $u_{i,14}^\kappa$ to $c(y_{2i-1}^\kappa) \wedge c(\alpha_{i,1}^\kappa) \wedge c(\alpha_{i,2}^\kappa)$ and $u_{i,13}^\kappa$ to the opposite. Similarly, it looks for all $m+1 \leq i \leq 2m$ at $y_{2i-1}^\kappa, \alpha_{i,1}^\kappa$, and $\alpha_{i,2}^\kappa$ and biases $u_{i,15}^\kappa$ to $c(y_{2i-1}^\kappa) \wedge (\neg c(\alpha_{i,1}^\kappa) \vee \neg c(\alpha_{i,2}^\kappa))$ and $u_{i,16}^\kappa$ to the opposite.

Type 14 looks at all nodes of type lower than 14 that are adjacent to g_i^κ with the single exception of η_i^κ if $g_i^\kappa = w_{j,k}^\kappa$. Namely, it looks at $u_{i,4j+2}^\kappa, u_{i,4j+3}^\kappa$ for $0 \leq j \leq 2$, at $\alpha_{j,k}^\kappa$ and $\sigma_{j,k}^\kappa$ if $I_k(g_j)^\kappa = (g_i^\kappa)$ for $k \in \{1, 2\}$, at $u_{i,13}^\kappa$ if $i \leq m$, at $u_{i,15}^\kappa$ if $m+1 \leq i \leq 2m$. Furthermore, it looks at μ_i^κ if $g_i^\kappa = w_{j,k}^\kappa$, at $\alpha_{i,1}^\kappa$, and $\alpha_{i,2}^\kappa$. The component treats the color of μ_i^κ as if it was the color of η_i^κ if $g_i^\kappa = w_{j,k}^\kappa$ – the component cannot look at η_i^κ since η_i^κ would have a degree of six in this case but we will see in Lemma 6 that $c(\eta_i^\kappa) = c(\mu_i^\kappa)$ in any local optimum. Then, the component computes whether g_i^κ is weakly indifferent and the color $\rho \in \{0, 1\}$ it would have if it was a single node and not weakly indifferent. It biases g_i^κ to ρ if it is not weakly indifferent. If g_i^κ is weakly indifferent then it biases g_i^κ to $c(\alpha_{i,1}^\kappa) \wedge c(\alpha_{i,2}^\kappa)$. This finishes the description of G_C .

Now we consider the colors of the nodes of G_C in an arbitrary local optimum. All of the remaining Lemmas have an inherent statement “for any local optimum P ”. We call a gate g_i^κ **correct** if $\text{col}(g_i^\kappa) = \neg(\text{col}(I_1(g_i^\kappa)) \vee \text{col}(I_2(g_i^\kappa)))$. In the following we will, among other things, argue about the colors of the comparing nodes $v \in V^C$ in P . We do this by naming the decisive neighbors of v , their colors, and the color to which v is biased. Then, we can deduce the color of v via Theorem 1 – recall that a necessary condition of Theorem 1 is that v is biased to the opposite color as the color of its decisive neighbors if v is not weakly indifferent. The following Lemmas characterize properties of some components.

Lemma 5. d^0, d^1 and g_i^κ for any $1 \leq i \leq N - n$, $\kappa \in \{0, 1\}$ are comparing nodes. Either $\text{col}(g_i^\kappa) = 1$ or $\text{col}(g_i^\kappa) = 0$ for all $1 \leq i \leq N$. Moreover, $c(u^0) \neq c(u^1)$.

Proof. In Table 1 we name all nodes adjacent to d^0, d^1 , and g_i^κ for all $1 \leq i \leq N - n$, $\kappa \in \{0, 1\}$ and the weights of the corresponding edges. By means of the table it can easily be verified that the aforementioned nodes are comparing.

Now consider the nodes g_i^κ . Recall first that Theorem 1 only applies to local optima in which the comparing node is biased to the color that it had if it was a single node. The only nodes different from the constants that are incident to any g_j^κ and to which the components of type 14 do not look at is μ_i^κ for $g_j^\kappa = w_{i,k}^\kappa$ and any $k \in \{1, 2\}$. From Lemma 6 we know that $c(\eta_i^\kappa) = c(\mu_i^\kappa)$. Thus, the component of type 14 correctly decides whether g_i^κ is weakly indifferent as outlined in the description of type 14 and therefore it biases g_i^κ such that Theorem 1 implies that either $\text{col}(g_i^\kappa) = 1$ or $\text{col}(g_i^\kappa) = 0$ for all $1 \leq i \leq N$.

Due to the weights of the edges incident to u^0 and u^1 and since they are biased to different colors by type 9 in each local optimum at least one of them is unhappy if both have the same color. Thus, the claim follows. \square

Lemma 6 (similar to Claims 5.9.B and 5.10.B in [22]). If $\text{col}(d^\kappa) = 1$ then neither flipping $w_{i,1}^\kappa$ nor $w_{i,2}^\kappa$ change the cut by a weight of type 7. If $\text{col}(d^\kappa) = 0$ and $\text{col}(w_{i,1}^\kappa) = \text{col}(w_{i,2}^\kappa)$ then $\text{col}(w_{i,1}^\kappa) \neq c(\eta_i^\kappa)$. Moreover, $c(\eta_i^\kappa) \neq c(\lambda_i^\kappa) \neq c(\mu_i^\kappa)$.

Proof. The proof uses the following claim.

Claim 3. If $\text{col}(d^\kappa) = \rho$ for $\rho \in \{0, 1\}$ then $c(\hat{d}_i^\kappa) = \bar{\rho}$ for all $1 \leq i \leq n$.

Proof. There are three edges incident to each node \hat{d}_i^κ as introduced in type 6. Namely, one edge of type 6 and two edges of type 7. Since the weight of the edge of type 6 is greater than the sum of all edges of higher type, in particular the two edges of type 7, the claim follows. \square

Assume $col(d^\kappa) = 1$. Then, by Claim 3 we have $c(\hat{d}_i^\kappa) = 0$ for all i . Since $col(d^\kappa) = 1$, the weights of the five edges incident to $\theta_{i,1}^\kappa$ as depicted in Figure 7 imply $c(\theta_{i,1}^\kappa) \neq c(\eta_i^\kappa)$. Similarly, we can argue that $c(\theta_{i,2}^\kappa) \neq c(\eta_i^\kappa)$. But then, neither a flip of $w_{i,1}^\kappa$ nor a flip of $w_{i,2}^\kappa$ can change the cut by a weight of type 7.

Now assume $col(d^\kappa) = 0$ and $col(w_{i,1}^\kappa) = col(w_{i,2}^\kappa)$. Due to Claim 3 we have $c(\hat{d}_i^\kappa) = 1$ for all i . The weights of the edges incident to $\theta_{i,1}^\kappa$ and $\theta_{i,2}^\kappa$ imply $c(\theta_{i,1}^\kappa) = 1$ and $c(\theta_{i,2}^\kappa) = 0$. Since $col(w_{i,1}^\kappa) = col(w_{i,2}^\kappa)$ and $c(\theta_{i,1}^\kappa) \neq c(\theta_{i,2}^\kappa)$, node η_i^κ is happy if and only if its color is different from the color of $w_{i,1}^\kappa$ and $w_{i,2}^\kappa$.

Finally, the claim $c(\eta_i^\kappa) \neq c(\lambda_i^\kappa) \neq c(\mu_i^\kappa)$ follows directly from the weights of the edges incident to λ_i^κ and μ_i^κ . \square

Lemma 7 (similar to Lemma 4.1H in [22]). *If $c(z_j^\kappa) = 1$ then $c(y_{j-1}^\kappa) = 0$. If $c(y_j^\kappa) = 0$ then $c(y_p^\kappa) = 0$ and $c(z_p^\kappa) = 1$ for all $p \leq j$.*

Proof. The sum of the weights of the edges $\{z_j^\kappa, y_{j-1}^\kappa\}$ and $\{y_{j-1}^\kappa, 1\}$ is greater than the sum of all other edges incident to y_{j-1}^κ . Thus, if $c(z_j^\kappa) = 1$ then $c(y_{j-1}^\kappa) = 0$. Similarly, we can argue that z_p^κ has its unnatural value if y_p^κ has its unnatural value. Therefore, the claim follows by induction. \square

Lemma 8 (similar to Lemma 4.1 in [22]). *If g_i^κ is not correct then $c(z_{2i}^\kappa) = 1$.*

Proof. The proof uses the following claims.

Claim 4. *If $c(z_{2i}^\kappa) = 0$ then $c(y_{2i-1}^\kappa) = 1$.*

Proof. Assume $c(z_{2i}^\kappa) = c(y_{2i-1}^\kappa) = 0$. If y_{2i-1}^κ is biased to black by the component of type 8 then $c(y_{2i-1}^\kappa) = 1$ since $c(z_{2i}^\kappa) = 0$ which is a contradiction. Thus, y_{2i-1}^κ is biased to 0. Since z_{2i}^κ and y_{2i-1}^κ are biased to opposite colors by type 8, node z_{2i}^κ is biased to 1. Due to the weight of its incident edges it cannot be white then. But this is a contradiction. \square

Node	Neighbor	Type	R. Weight	Condition
d^κ	1 u^κ	2	1	
	0 y_0^κ	4	2^3	
	1 z_0^κ		2^1	
	$u_{i,14}^\kappa$ $u_{i,16}^\kappa$	5	2^{2i}	$1 \leq i \leq m$
	κ κ		1	
	1 \hat{d}_i^κ	6	2^{2i}	$1 \leq i \leq n$
	$\theta_{i,1}^\kappa$ 0	7	2^{2i}	$1 \leq i \leq m$
	no name	10	1	

Node	Condition	Neighbor	Type	R. Weight	Condition
g_i^κ	$1 \leq i \leq N - n$	$u_{i,2}$ $u_{i,3}$	3	2^{10i+5}	
		$u_{i,10}$ $u_{i,11}$		2^{10i+1}	
		$u_{i,6}$ $u_{i,7}$		2^{10i-1}	
		1 $\alpha_{j,1}^\kappa$		2^{10j+7}	$I_1(g_j) = g_i$
	$2(n+m)+1 \leq i \leq N - n$	0 $\sigma_{j,1}^\kappa$	3	2^{10j+5}	
		0 $\sigma_{j,2}^\kappa$		2^{10j+5}	$I_2(g_j) = g_i$
		1 $\alpha_{j,2}^\kappa$		2^{10j-1}	
		0 $u_{i,13}^\kappa$	5	2^{2i}	
	$1 \leq i \leq m$	1 $u_{i,15}^\kappa$			
	$m+1 \leq i \leq 2m$	$\theta_{j,2}^\kappa$ η_j^κ	7	2	$j := i - 2m - n$
	$2m+n+1 \leq i \leq 2(m+n)$	$\theta_{j,1}^\kappa$ η_j^κ			
	$2m+1 \leq i \leq 2m+n$	$\theta_{j,1}^\kappa$ η_j^κ			
		no name	14	1	

Table 1. Neighborhood of the nodes d^0, d^1 , and g_i^κ for $1 \leq i \leq N - n$, $\kappa \in \{0, 1\}$.

Claim 5. If $\text{col}(I_1(g_i^\kappa)) = 1$ then $c(\alpha_{i,1}^\kappa) = 0$ and $c(\beta_{i,1}^\kappa) = 1$. If $\text{col}(I_2(g_i^\kappa)) = 1$ then $c(\alpha_{i,2}^\kappa) = 0$ and $c(\beta_{i,2}^\kappa) = 1$.

Proof. If $\text{col}(I_1(g_i^\kappa)) = 1$ then $c(\alpha_{i,1}^\kappa) = 0$ since the edges $\{I_1(g_i^\kappa), \alpha_{i,1}^\kappa\}$ and $\{\alpha_{i,1}^\kappa, 1\}$ combined weigh more than the sum of all other edges incident to $\alpha_{i,1}^\kappa$. Analogously, $c(\alpha_{i,1}^\kappa) = 0$ implies $c(\beta_{i,1}^\kappa) = 1$. The argumentation for the second part of the claim is similar. \square

Claim 6. If $\text{col}(I_1(g_i^\kappa)) = 1$ and $\text{col}(g_i^\kappa) = 1$ then $c(z_{2i}^\kappa) = 1$. If $\text{col}(I_2(g_i^\kappa)) = 1$ and $\text{col}(g_i^\kappa) = 1$ then $c(z_{2i}^\kappa) = 1$.

Proof. Assume for the sake of contradiction $\text{col}(I_1(g_i^\kappa)) = 1$, $\text{col}(g_i^\kappa) = 1$, but $c(z_{2i}^\kappa) = 0$. Claim 4 implies $c(y_{2i-1}^\kappa) = 1$ since $c(z_{2i}^\kappa) = 0$. Moreover, Lemma 7 implies $c(y_{2i+1}^\kappa) = 1$ since $c(z_{2i}^\kappa) = 0$. Thus, $c(y_{2i+1}^\kappa) = c(y_{2i-1}^\kappa)$ and therefore the nodes $u_{i,3}^\kappa$ and $u_{i,4}^\kappa$ are biased to 0 and 1 respectively by the component of type 12. Then, $c(u_{i,3}^\kappa) = 0$ and therefore $c(u_{i,4}^\kappa) = 1$. Then, Claim 5 implies $c(\gamma_{i,1}^\kappa) = 0$ and thereafter $c(z_{2i+1}^\kappa) = 1$. Then, $c(z_{2i}^\kappa) = 1$ due to Lemma 7 which is a contradiction. The proof for $\text{col}(I_2(g_i^\kappa)) = 1$ is analogous. \square

Claim 7. If $\text{col}(I_1(g_i^\kappa)) = 0$ then $c(\delta_{i,1}^\kappa) = 1$. If $\text{col}(I_2(g_i^\kappa)) = 0$ then $c(\delta_{i,2}^\kappa) = 1$.

Proof. If $\text{col}(I_1(g_i^\kappa)) = 0$ then $c(\sigma_{i,1}^\kappa) = 1$ since the edges $\{I_1(g_i^\kappa), \sigma_{i,1}^\kappa\}$ and $\{\sigma_{i,1}^\kappa, 0\}$ combined weigh more than the sum of all other edges incident to $\sigma_{i,1}^\kappa$. Similarly, it follows that $c(\tau_{i,1}^\kappa) = 0$ and $c(\delta_{i,1}^\kappa) = 1$. Analogously, $\text{col}(I_2(g_i^\kappa)) = 0$ implies $c(\delta_{i,2}^\kappa) = 1$. \square

Claim 8. If $\text{col}(I_1(g_i^\kappa)) = \text{col}(I_2(g_i^\kappa)) = 0$ then $c(\beta_{i,3}^\kappa) = 0$.

Proof. Due to Claim 7, $c(\delta_{i,1}^\kappa) = c(\delta_{i,2}^\kappa) = 1$. Since the sum of the weights of the edges $\{\beta_{i,3}^\kappa, \delta_{i,1}^\kappa\}$ and $\{\beta_{i,3}^\kappa, \delta_{i,2}^\kappa\}$ is greater than the sum of all other edges incident to $\beta_{i,3}^\kappa$ the claim follows. \square

Claim 9. If $\text{col}(I_1(g_i^\kappa)) = \text{col}(I_2(g_i^\kappa)) = \text{col}(g_i^\kappa) = 0$ then $c(z_{2i}^\kappa) = 1$.

Proof. Assume $\text{col}(I_1(g_i^\kappa)) = \text{col}(I_2(g_i^\kappa)) = \text{col}(g_i^\kappa) = 0$ but $c(z_{2i}^\kappa) = 0$. Then, Lemma 7 implies $c(y_{2i+1}^\kappa) = 1$ since $c(z_{2i}^\kappa) = 0$. Moreover, Claim 4 implies $c(y_{2i-1}^\kappa) = 1$. Thus, $c(y_{2i+1}^\kappa) = c(y_{2i-1}^\kappa)$ and therefore the nodes $u_{i,11}^\kappa$ and $u_{i,12}^\kappa$ are biased to 1 and 0 respectively by the component of type 12. Then, $c(u_{i,11}^\kappa) = 1$ and therefore $c(u_{i,12}^\kappa) = 0$. But then Claim 8 implies $c(\gamma_{i,3}^\kappa) = 1$ and therefore $c(y_{2i}^\kappa) = 0$. Then, $c(z_{2i}^\kappa) = 1$ due to Lemma 7 which is a contradiction. Thus, the claim follows. \square

Lemma 5 implies that either $\text{col}(g_i^\kappa) = 1$ or $\text{col}(g_i^\kappa) = 0$. If $\text{col}(I_1(g_i^\kappa)) = 1$ or $\text{col}(I_2(g_i^\kappa)) = 1$ then the claim follows from Claim 6. If $\text{col}(I_1(g_i^\kappa)) = \text{col}(I_2(g_i^\kappa)) = 0$ then the claim follows from Claim 9. \square

Lemma 9 (partially similar to Lemma 4.2 in [22]). If $c(y_{2i+1}^\kappa) = 0$ then $c(\alpha_{i,1}^\kappa) = c(\alpha_{i,2}^\kappa) = 0$ and $c(\sigma_{i,1}^\kappa) = c(\sigma_{i,2}^\kappa) = 1$.

Proof. Assume $c(y_{2i+1}^\kappa) = 0$. From Lemma 7 we know that $c(z_{2i+1}^\kappa) = c(z_{2i}^\kappa) = 1$ and $c(y_{2i}^\kappa) = c(y_{2i-1}^\kappa) = 0$. We proof is done by means of the following two claims.

Claim 10. Assume $c(y_{2i+1}^\kappa) = 0$. Then, $c(\alpha_{i,1}^\kappa) = c(\alpha_{i,2}^\kappa) = 0$, $c(\beta_{i,1}^\kappa) = c(\beta_{i,2}^\kappa) = 1$, and $c(\gamma_{i,1}^\kappa) = c(\gamma_{i,2}^\kappa) = 0$.

Proof. From Lemma 5 we know that either $\text{col}(g_i^\kappa) = 1$ or $\text{col}(g_i^\kappa) = 0$. From the component of type 11 node $\beta_{i,1}^\kappa$ is biased to 1 and $\gamma_{i,1}^\kappa$ is biased to 0.

Assume first $c(\beta_{i,1}^\kappa) = c(\gamma_{i,1}^\kappa) = 1$. Then, node $\gamma_{i,1}^\kappa$ is unhappy since $c(z_{2i+1}^\kappa) = 1$ due to Lemma 7. Now assume $c(\beta_{i,1}^\kappa) = c(\gamma_{i,1}^\kappa) = 0$. Then, node $\beta_{i,1}^\kappa$ is unhappy. Now assume $c(\beta_{i,1}^\kappa) = 0$ and $c(\gamma_{i,1}^\kappa) = 1$. If $\text{col}(g_i^\kappa) = 0$ then $c(u_{i,2}^\kappa) = 1$ and $c(u_{i,1}^\kappa) = 0$ due to their bias from the component of type 12 – recall that $c(y_{2i+1}^\kappa) = c(y_{2i-1}^\kappa)$ due to Lemma 7. But then $\beta_{i,1}^\kappa$ is unhappy, which is a contradiction. Now assume $\text{col}(g_i^\kappa) = 1$. Then, $c(u_{i,3}^\kappa) = 0$ and $c(u_{i,4}^\kappa) = 1$ due to the bias of type 12. But then $\gamma_{i,1}^\kappa$ is unhappy since $c(z_{2i+1}^\kappa) = 1$ due to Lemma 7 which is also a contradiction. Thus, $c(\beta_{i,1}^\kappa) = 1$ and $c(\gamma_{i,1}^\kappa) = 0$.

Since $c(\beta_{i,1}^\kappa) = 1$ node $\alpha_{i,1}^\kappa$ must be white since it is biased to white by type 11. The proof for $\alpha_{i,1}^\kappa, \beta_{i,2}^\kappa$, and $\gamma_{i,2}^\kappa$ is analogous. \square

Claim 11. Assume $c(y_{2i+1}^\kappa) = 0$. Then, $c(\sigma_{i,1}^\kappa) = c(\sigma_{i,2}^\kappa) = 1$, $c(\beta_{i,3}^\kappa) = 0$, and $c(\gamma_{i,3}^\kappa) = 1$.

Proof. Assume first that $c(\delta_{i,1}^\kappa) = c(\delta_{i,2}^\kappa) = 0$. Type 11 biases $\delta_{i,1}^\kappa$ and $\delta_{i,2}^\kappa$ to black. Therefore, both nodes $\delta_{i,1}^\kappa$ and $\delta_{i,2}^\kappa$ are unhappy. Therefore, we may assume that at least one of them is black.

If $c(\beta_{i,3}^\kappa) = c(\gamma_{i,3}^\kappa) = 1$ then $\beta_{i,3}^\kappa$ is unhappy because $\beta_{i,3}^\kappa$ is biased to 0 by type 11. Now assume $c(\beta_{i,3}^\kappa) = c(\gamma_{i,3}^\kappa) = 0$. Then, node $\gamma_{i,3}^\kappa$ is unhappy since $c(y_{2i}) = 0$ has its unnatural value due to Lemma 7 and since $\gamma_{i,3}^\kappa$ is biased to 1 by type 11. Now assume $c(\beta_{i,3}^\kappa) = 1$ and $c(\gamma_{i,3}^\kappa) = 0$. If $col(g_i^\kappa) = 0$ then the bias of type 12 implies $c(u_{i,11}^\kappa) = 1$ and $c(u_{i,12}^\kappa) = 0$ which is a contradiction since $\gamma_{i,3}^\kappa$ is unhappy then due to the bias of type 11. But if $col(g_i^\kappa) = 1$ then the bias of type 12 implies $c(u_{i,10}^\kappa) = 0$ and $c(u_{i,9}^\kappa) = 1$ which is also a contradiction since $\beta_{i,3}^\kappa$ is unhappy then due to the bias of type 11. Thus, $c(\beta_{i,3}^\kappa) = 0$ and $c(\gamma_{i,3}^\kappa) = 1$.

Since $c(\beta_{i,3}^\kappa) = 0$ we get $c(\delta_{i,1}^\kappa) = c(\delta_{i,2}^\kappa) = 1$ due to the biases of type 11. Then, $c(\tau_{i,1}^\kappa) = c(\tau_{i,2}^\kappa) = 0$ and therefore $c(\sigma_{i,1}^\kappa) = c(\sigma_{i,2}^\kappa) = 1$ also due to the biases of type 11. \square

Lemma 10 (partially similar to Lemma 4.3 in [22]). *Assume $c(y_{2i+1}^\kappa) = 1$ and $c(y_{2i-1}^\kappa) = 0$. If g_i^κ is correct then z_{2i}^κ , z_{2i+1}^κ , and y_{2i}^κ have the colors to which they are biased by type 8. If g_i^κ is not correct then flipping g_i^κ does not decrease the cut by a weight of an edge type 3 corresponding to g_i^κ and increases it by a weight of type 14 if g_i is indifferent with respect to edges of type 5 and 7.*

Proof. The proof uses the following three claims.

Claim 12. *Assume $c(y_{2i+1}^\kappa) = 1$. Then, $c(\alpha_{i,1}^\kappa) = \neg col(I_1(g_i^\kappa))$ and $c(\alpha_{i,2}^\kappa) = \neg col(I_2(g_i^\kappa))$. If, in addition, $c(y_{2i-1}^\kappa) = 0$ then $c(\beta_{i,1}^\kappa) = col(I_1(g_i^\kappa))$ and $c(\beta_{i,2}^\kappa) = col(I_2(g_i^\kappa))$.*

Proof. If $col(I_1(g_i^\kappa)) = 1$ then $c(\alpha_{i,1}^\kappa) = 0$. If, on the other hand, $col(I_1(g_i^\kappa)) = 0$ then $c(\alpha_{i,1}^\kappa) = 1$ since $\alpha_{i,1}^\kappa$ is biased to 1 by type 11. Now assume $c(y_{2i-1}^\kappa) = 0$. If $col(I_1(g_i^\kappa)) = 1$ then $c(\beta_{i,1}^\kappa) = 1$ since $c(\alpha_{i,1}^\kappa) = 0$. Now assume $col(I_1(g_i^\kappa)) = 0$. Due to $c(\alpha_{i,1}^\kappa) = 1$ and since $\beta_{i,1}^\kappa$ is biased to 0 by type 11, it can only be black if $\gamma_{i,1}^\kappa$ and $u_{i,1}^\kappa$ are both white. But if $\gamma_{i,1}^\kappa$ is white then $u_{i,4}^\kappa$ must be black since $\gamma_{i,1}^\kappa$ is biased to black by type 11. If $col(g_i^\kappa) = 1$ then $c(u_{i,2}^\kappa) = 0$ and $c(u_{i,1}^\kappa) = 1$ due to the bias of type 12 which is a contradiction. On the other hand, if $col(g_i^\kappa) = 0$ then $c(u_{i,3}^\kappa) = 1$ and $c(u_{i,4}^\kappa) = 0$ due to the bias of type 12 which is also a contradiction. Thus, $c(\beta_{i,1}^\kappa) = 0$.

The argumentation for $\alpha_{i,2}^\kappa$ and $\beta_{i,2}^\kappa$ is analogous. \square

Claim 13. *Assume $c(y_{2i+1}^\kappa) = 1$ and $c(y_{2i-1}^\kappa) = 0$. Then, $c(\beta_{i,3}^\kappa) = col(I_1(g_i^\kappa)) \vee col(I_2(g_i^\kappa))$.*

Proof. If an input is white then the corresponding $\delta_{i,j}^\kappa$ is black due to Claim 7. Thus, if both inputs are white then $\beta_{i,3}^\kappa$ is white.

Now assume that at least one input is black. Let $I_1(g_i^\kappa) = 1$. Since $\sigma_{i,1}^\kappa$ is biased to white, we have $c(\sigma_{i,1}^\kappa) = 0$. Analogously, we get $c(\tau_{i,1}^\kappa) = 1$. Node $\delta_{i,1}^\kappa$ is biased to white by type 11. If both nodes $\delta_{i,1}^\kappa$ and $\delta_{i,2}^\kappa$ are black then $\delta_{i,1}^\kappa$ is unhappy. Thus, we may assume that at least one of them is white. Since $\beta_{i,3}^\kappa$ is biased to 1 by type 11, it can only be white if $\gamma_{i,3}^\kappa$ and $u_{i,9}^\kappa$ are both black. But if $\gamma_{i,3}^\kappa$ is black then $u_{i,12}^\kappa$ must be white since $\gamma_{i,3}^\kappa$ is biased to white by type 11. Then, the bias of type 12 implies that if g_i^κ is white then $u_{i,10}^\kappa$ is black and $u_{i,9}^\kappa$ is white and if g_i^κ is black then $u_{i,11}^\kappa$ is white and $u_{i,12}^\kappa$ is black, each resulting in a contradiction. Thus, $c(\beta_{i,3}^\kappa) = 1$. \square

Claim 14. *Assume $c(y_{2i+1}^\kappa) = 1$ and $c(y_{2i-1}^\kappa) = 0$. If g_i^κ is correct then $c(\gamma_{i,1}^\kappa) = c(\gamma_{i,2}^\kappa) = 1$ and $c(\gamma_{i,3}^\kappa) = 0$. If g_i^κ is not correct then at least one of the nodes $c(u_{i,2}^\kappa)$, $c(u_{i,3}^\kappa)$ has the same color as g_i^κ , at least one of the nodes $c(u_{i,6}^\kappa)$, $c(u_{i,7}^\kappa)$ has the same color as g_i^κ , and at least one of the nodes $c(u_{i,10}^\kappa)$, $c(u_{i,11}^\kappa)$ has the same color as g_i^κ .*

Proof. Assume first that g_i^κ is correct. From Claim 12 we know that $c(\beta_{i,1}^\kappa) = col(I_1(g_i^\kappa))$. Since g_i^κ is correct, at least one of the two nodes $\beta_{i,1}^\kappa$ and g_i^κ is white. Assume first that $c(\beta_{i,1}^\kappa) = 1$. Then, due to Claim 12, we have $c(\alpha_{i,1}^\kappa) = 0$. If g_i^κ is white then $c(u_{i,3}^\kappa) = 1$ and $c(u_{i,4}^\kappa) = 0$ since they are biased to 1 and 0 respectively by type 12. Since at least one of the nodes $u_{i,4}^\kappa$ and $\beta_{i,1}^\kappa$ is white and $\gamma_{i,1}^\kappa$ is biased to black by type 11 it is actually black. Analogously, we can argue that $\gamma_{i,2}^\kappa$ is also black. Moreover, by Claim 13 we know that $c(\beta_{i,3}^\kappa) = col(I_1(g_i^\kappa)) \vee col(I_2(g_i^\kappa))$. Since g_i^κ is correct, it has the opposite color as $\beta_{i,3}^\kappa$. If $col(g_i^\kappa) = 1$ then $c(\alpha_{i,1}^\kappa) = c(\alpha_{i,2}^\kappa) = 1$ and therefore $c(u_{i,11}^\kappa) = 0$ and $c(u_{i,12}^\kappa) = 1$ since they are biased to 0 and 1 respectively by type 12. Therefore, at least one of the nodes $u_{i,12}^\kappa$ and $\beta_{i,3}^\kappa$ is black. Thus, $\gamma_{i,3}^\kappa$ has the color to which it is biased by type 11, i. e. 0.

Now assume that g_i^κ is not correct. If $\text{col}(I_1(g_i^\kappa)) = 1$ then $c(\alpha_{i,1}^\kappa) = 0$ and $c(\beta_{i,1}^\kappa) = 1$ due to Claim 12. Moreover, since g_i^κ is not correct, we have $\text{col}(g_i^\kappa) = 1$. Then $c(\alpha_{i,1}^\kappa) = 0$ and the biases of type 12 imply $c(u_{i,1}^\kappa) = 0$ and $c(u_{i,2}^\kappa) = 1$. If $\text{col}(I_1(g_i^\kappa)) = 0$ then $c(\alpha_{i,1}^\kappa) = 1$ and $c(\beta_{i,1}^\kappa) = 0$ due to Claim 12. Since $\gamma_{i,1}^\kappa$ is biased to 1 by type 11 we get $c(\gamma_{i,1}^\kappa) = 1$. Moreover, since $c(\alpha_{i,1}^\kappa) = 1$ the biases of type 12 imply $c(u_{i,1}^\kappa) = 1$, $c(u_{i,2}^\kappa) = 0$, $c(u_{i,4}^\kappa) = 0$ and $c(u_{i,3}^\kappa) = 1$. The proof for $c(u_{i,6}^\kappa)$ and $c(u_{i,7}^\kappa)$ is analogous. By Claim 13 we know that $c(\beta_{i,3}^\kappa) = \text{col}(I_1(g_i^\kappa)) \vee \text{col}(I_2(g_i^\kappa))$. Since g_i^κ is not correct, we have $\text{col}(g_i^\kappa) = c(\beta_{i,3}^\kappa)$. If $c(\beta_{i,3}^\kappa) = 0$ then, due to Claim 12 we have $c(\alpha_{i,1}^\kappa) = c(\alpha_{i,2}^\kappa) = 1$. Then, the biases of the component of type 12 imply $c(u_{i,9}^\kappa) = 1$ and $c(u_{i,10}^\kappa) = 0$. Thus, $u_{i,10}^\kappa$ has the same color as g_i^κ . If $c(\beta_{i,3}^\kappa) = 1$ then $c(\gamma_{i,3}^\kappa) = 0$ since it is biased to white by type 11. Moreover, $c(\alpha_{i,1}^\kappa) = 0$ or $c(\alpha_{i,2}^\kappa) = 0$ due to Claim 12. Then, the biases of the component of type 12 imply $c(u_{i,9}^\kappa) = 0$ and $c(u_{i,10}^\kappa) = 1$ as well as $c(u_{i,12}^\kappa) = 1$ and $c(u_{i,11}^\kappa) = 0$. Then, we have $c(u_{i,10}^\kappa) \neq c(u_{i,11}^\kappa)$ which proves the claim. \square

Assume $c(y_{2i+1}^\kappa) = 1$ and $c(y_{2i-1}^\kappa) = 0$. Assume furthermore that g_i^κ is correct. Then, due to Claim 14 we have $c(\gamma_{i,1}^\kappa) = c(\gamma_{i,2}^\kappa) = 1$ and $c(\gamma_{i,3}^\kappa) = 0$. Then, if the nodes y_j^κ, z_j^κ for all j are biased to their natural values then due to $c(y_{2i+1}^\kappa) = 1$ we get $c(z_{2i+1}^\kappa) = 0$, $c(y_{2i}^\kappa) = 1$, and $c(z_{2i}^\kappa) = 0$. If, on the other hand, the nodes y_j^κ, z_j^κ for all j are biased to their unnatural values then due to $c(y_{2i-1}^\kappa) = 0$ we get $c(z_{2i}^\kappa) = 1$, $c(y_{2i}^\kappa) = 0$, and $c(z_{2i+1}^\kappa) = 1$.

Now assume that g_i^κ is not correct. Due to $c(y_{2i-1}^\kappa) = 0$ Lemma 7 implies $c(y_{2j+1}^\kappa) = 0$ for all $j < i$. Then, Lemma 9 implies $c(\alpha_{j,1}^\kappa) = c(\alpha_{j,2}^\kappa) = 0$ and $c(\sigma_{j,1}^\kappa) = c(\sigma_{j,2}^\kappa) = 1$ for all $j < i$. Then, Claim 14 implies that flipping g_i^κ does not decrease the cut by a weight of type 3. Finally, Claim 12 implies $c(\alpha_{i,j}^\kappa) = \neg \text{col}(I_j(g_i^\kappa))$ for $1 \leq j \leq 2$. Thus, flipping g_i^κ to its correct color gains a weight of type 14 if g_i^κ is indifferent with respect to edges of type 5 and 7. \square

Lemma 11. *If $\text{col}(d^\kappa) = 1$, $\text{col}(d^\kappa) = 0$, and all nodes y_i^κ, z_i^κ for $0 \leq i \leq 2N+1$ are biased to their natural values then $c(y_1^\kappa) = 1$.*

Proof. Assume $\text{col}(d^\kappa) = 1$, $\text{col}(d^\kappa) = 0$, and that all nodes y_i^κ, z_i^κ for $0 \leq i \leq 2N+1$ are biased to their natural values. We show that all gates of G_C^κ are correct. For the sake of contradiction we assume that G_C^κ contains an incorrect gate and let g_i^κ be the incorrect gate with the highest index.

We first show by induction that the nodes y_j^κ, z_j^κ for $j > 2i+1$ and y_{2i+1}^κ have their natural values. Since y_{2N+1}^κ is biased to its natural value, we have $c(y_{2N+1}^\kappa) = 1$. Assume $c(y_{2j+1}^\kappa) = 1$ for any $j > i$. If any one of the nodes $z_{2j+1}^\kappa, y_{2j}^\kappa, z_{2j}^\kappa$ has its unnatural value then Lemma 7 implies $c(y_{2j-1}^\kappa) = 0$. Then, Lemma 10 implies that all nodes $z_{2j+1}^\kappa, y_{2j}^\kappa, z_{2j}^\kappa$ have their natural values whereafter Claim 4 implies $c(y_{2i-1}^\kappa) = 1$ which is a contradiction. Thus, $c(y_{2j+1}^\kappa) = 1$ implies $c(y_{2j-1}^\kappa) = 1$ for any $j > i$ and therefore it follows by induction that all nodes y_j^κ, z_j^κ for $j > 2i+1$ and y_{2i+1}^κ have their natural values.

Since g_i^κ is incorrect, all nodes y_j^κ, z_j^κ for $j \leq 2i-1$ have their unnatural values due to Lemma 8 and 7. According to Lemma 9 and 10 correcting g_i^κ does not decrease the cut by a weight of type 3 and gains a weight of type 14. In the following, we distinguish between three cases for the index i and show that g_i^κ is unhappy in each of the cases. First, if $i > 2n+2m$ then there are no node edges of type 5 or 7 incident to g_i^κ . Thus, g_i^κ is unhappy then. Second, if $2m+1 \leq i \leq 2n+2m$ then there are no edges of type 5 incident to g_i^κ . Due to Lemma 6 correcting g_i^κ does not decrease the cut by a weight of type 7. Third, if $i \leq 2m$ then there are no edges of type 7 incident to g_i^κ . Correcting g_i^κ does not decrease the cut by a weight of type 5 since due to the biases of type 13 we have $c(u_{i,14}) = 0$, $c(u_{i,13}) = 1$ for $i \leq m$ and $c(u_{i,16}) = 1$, $c(u_{i,15}) = 0$ for $m < i \leq 2m$. Altogether, g_i^κ is unhappy in each of the three cases which is a contradiction. Thus, g_i^κ is correct for all i . Thus, all nodes y_i^κ, z_i^κ for $1 \leq i \leq 2N+1$ have their natural values.

Lemma 12. *If $c(y_1^\kappa) = c(u^\kappa) = 0$ and $c(u^\kappa) = 1$ then $\text{col}(d^\kappa) = 1$ and $\text{col}(d^\kappa) = 0$.*

Proof. Assume $c(y_1^\kappa) = c(u^\kappa) = 0$ and $c(u^\kappa) = 1$. Then, independently of the color of y_1^κ , node d^κ is biased to 1 and d^κ to 0 by type 10 – recall that Theorem 1 only applies to local optima in which the comparing node is biased to the color that it had if it was a single node. Lemma 7 implies $c(y_0^\kappa) = 0$. Since $c(u^\kappa) = 0$ and $c(y_0^\kappa) = 0$ node y_0^κ and its counterpart, namely the constant 0, are decisive for d^κ . Thus, Theorem 1 implies $\text{col}(d^\kappa) = 1$.

Since $c(u^\kappa) = 1$, node u^κ and its counterpart, namely the constant 1, are decisive for d^κ . Thus, Theorem 1 implies $\text{col}(d^\kappa) = 0$. \square

Lemma 13. *If $c(y_1^\kappa) = c(u^\kappa) = 0$ and $c(y_1^{\bar{\kappa}}) = c(u^{\bar{\kappa}}) = c(y_0^{\bar{\kappa}}) = 1$ then $col(d^{\bar{\kappa}}) = 0$.*

Proof. Assume $c(y_1^\kappa) = c(u^\kappa) = 0$ and $c(y_1^{\bar{\kappa}}) = c(u^{\bar{\kappa}}) = c(y_0^{\bar{\kappa}}) = 1$. Then, Lemma 7 implies $c(z_0^\kappa) = 1$ since $c(y_1^\kappa) = 0$. Node $d^{\bar{\kappa}}$ is biased to 0 by type 10. Since $c(u^{\bar{\kappa}}) = 0$, $c(y_0^{\bar{\kappa}}) = 1$, and $c(z_0^\kappa) = 1$ node z_0^κ and its counterpart, i. e. the constant 1, are decisive for $d^{\bar{\kappa}}$. Thus, Theorem 1 implies $c(d^{\bar{\kappa}}) = 0$. \square

Lemma 14. *If $col(d^\kappa) = 1$, $col(d^{\bar{\kappa}}) = 0$, and all y_i^κ, z_i^κ are biased to their unnatural values by type 8 then they have their unnatural values.*

Proof. Assume that $col(d^\kappa) = 1$, $col(d^{\bar{\kappa}}) = 0$, and all y_i^κ, z_i^κ are biased to their unnatural values by type 8. Then, $col(d^{\bar{\kappa}}) = 0$ together with the bias to the unnatural value imply $c(z_0^\kappa) = 1$. Then, $col(d^{\bar{\kappa}}) = 1$ together with the bias to the unnatural value imply $c(y_0^\kappa) = 0$. Then, $c(z_1^\kappa) = 1$ and therefore $c(y_1^\kappa) = 0$. If $c(y_{j-1}^\kappa) = 0$ for any $2 \leq j \leq 2N+1$ then the bias to the unnatural value implies $c(z_j^\kappa) = 1$. Analogously, if $c(z_j^\kappa) = 1$ for any $2 \leq j \leq 2N+1$ then $c(y_j^\kappa) = 0$. Thus, the claim follows by induction. \square

Observation 15. *If $c(y_1^\kappa) = 1$ and all y_i^κ, z_i^κ are biased to their natural values by type 8 then $c(z_1^\kappa) = c(z_0^\kappa) = 0$ and $c(y_0^\kappa) = 1$.*

Lemma 15. *If $col(d^{\bar{\kappa}}) = 1$, $col(d^\kappa) = 0$ and all nodes $y_i^{\bar{\kappa}}, z_i^{\bar{\kappa}}$ are biased to their unnatural values by type 8 then $c(x^{\bar{\kappa}}) = c(\lambda^{\bar{\kappa}})$.*

Proof. Assume $col(d^{\bar{\kappa}}) = 1$, $col(d^\kappa) = 0$ and that all nodes $y_i^{\bar{\kappa}}, z_i^{\bar{\kappa}}$ are biased to their unnatural values by type 8. Then, Lemma 14 implies that $c(y_{2i+1}^{\bar{\kappa}}) = 0$ for each $0 \leq i \leq N$. Then, Lemma 6 implies $c(\lambda_i^{\bar{\kappa}}) \neq c(\mu_i^{\bar{\kappa}})$ for all $1 \leq i \leq n$. Thus, Lemma 9 implies that $x_i^{\bar{\kappa}}$ takes the colors of $\lambda_i^{\bar{\kappa}}$ for any $1 \leq i \leq n$. \square

Lemma 16. *Assume $c(y_1^\kappa) = col(d^{\bar{\kappa}}) = 1$ and $col(d^\kappa) = 0$. If $\kappa = 0$ and $C(x^0) \geq C(x^1)$ or $\kappa = 1$ and $C(x^1) > C(x^0)$ then $x^\kappa = w(x^\kappa)$.*

Proof. Assume $c(y_1^\kappa) = col(d^{\bar{\kappa}}) = 1$ and $col(d^\kappa) = 0$. We first consider the case that $\kappa = 0$ and $C(x^0) \geq C(x^1)$. Due to $C(x^0) \geq C(x^1)$ all y_i^0, z_i^0 are biased to their natural values by type 8. Since $c(y_1^0) = 1$ Lemma 8 and 7 together imply that all gates in G_C^0 compute correctly. Since all gates compute correctly we have $col(w_{i,1}^0) = col(w_{i,2}^0)$ for all $1 \leq i \leq n$. Then, Lemma 6 implies $col(w_{i,1}^0) = c(\lambda_i^1)$ for all i and therefore $w(x^0) = c(\lambda^1)$. Moreover, due to the same Lemma we also have $col(w_{i,1}^0) \neq c(\mu_i^1)$ for all i . Assume for the sake of contradiction, $c(x^1) \neq w(x^0)$. Then all nodes y_i^1, z_i^1 are biased to their unnatural values by type 8. Then, Lemma 14 implies that they have their unnatural values. Then, Lemma 9 implies that a flip of a node x_i^1 for any i does not decrease the cut by a weight of type 3. Thus the nodes x^1 assume the colors such that $x^1 = w(x^0)$ which is a contradiction. Thus, $x^1 = w(x^0)$. The case $\kappa = 1$ and $C(x^1) > C(x^0)$ is symmetric with the single difference that the case for the equality of $C(x^0)$ and $C(x^1)$ is obsolete. \square

Lemma 17. *Assume $c(y_{2i-1}^\kappa) = 1$, and $c(\alpha_{i,j}^\kappa) \neq col(I_j(g_i^\kappa))$ for $1 \leq j \leq 2$. Then, $c(u_{i,13}^\kappa) = c(u_{i,15}^\kappa) \neq col(g_i^\kappa)$ and $c(u_{i,14}^\kappa) = c(u_{i,16}^\kappa) = col(g_i^\kappa)$.*

Proof. Assume $c(y_{2i-1}^\kappa) = 1$, and $c(\alpha_{i,j}^\kappa) \neq col(I_j(g_i^\kappa))$ for $1 \leq j \leq 2$. Then, g_i^κ is correct due to Lemma 8 and 7. Since $c(y_{2i-1}^\kappa) = 1$, node u_i^{14} is biased to $c(\alpha_{i,1}^\kappa) \wedge c(\alpha_{i,2}^\kappa)$ by type 13. Thus, it is biased to black if and only if $\alpha_{i,1}^\kappa$ and $\alpha_{i,2}^\kappa$ are both black. But since $c(\alpha_{i,j}^\kappa) \neq col(I_j(g_i^\kappa))$ and $col(g_i^\kappa) = \neg(col(I_1(g_i^\kappa)) \vee col(I_2(g_i^\kappa)))$ node g_i^κ is also black if and only if $\alpha_{i,1}^\kappa$ and $\alpha_{i,2}^\kappa$ are both black. Thus, u_i^{14} is biased to the color that g_i^κ has and u_i^{13} to the opposite. Therefore, $c(u_{i,13}^\kappa) \neq c(u_{i,14}^\kappa) = col(g_i^\kappa)$.

Since $c(y_{2i-1}^\kappa) = 1$, node u_i^{15} is biased to $\neg c(\alpha_{i,1}^\kappa) \vee \neg c(\alpha_{i,2}^\kappa)$ by type 13. Therefore, it is biased to white if and only if $\alpha_{i,1}^\kappa$ and $\alpha_{i,2}^\kappa$ are both black. As in the previous case, it follows that u_i^{15} is biased to the opposite color that \hat{g}_i^κ has. Since u_i^{16} is biased to the opposite color, we have $c(u_{i,15}^\kappa) \neq c(u_{i,16}^\kappa) = col(\hat{g}_i^\kappa)$ which concludes the proof. \square

Now we continue to prove Theorem 2. Let P be a local optimum in G_C and assume w. l. o. g. that $C(x^0) \geq C(x^1)$. Then, all nodes y_i^0, z_i^0 are biased to their natural values by type 8. From Lemma 5 we know that $c(u^0) \neq c(u^1)$. In the following, we consider the four possible cases for the vector $c(y_1^0, y_1^1)$ and distinguish within each of these cases between the two cases for $c(u^0, u^1)$, if necessary. For the three cases

for the colors of $c(y_1^0, y_1^1)$ in which at least one node is white we show that they cannot occur in local optima and for the case that both nodes are black we show that the colors of the input nodes x^0 induce a local optimum of C .

(0, 0): Due to Lemma 7 we have $c(y_0^0) = c(y_0^1) = 0$ and $c(z_0^0) = c(z_0^1) = 1$. If $c(u^0, u^1) = (0, 1)$ then Lemma 12 implies $col(d^0) = 1$ and $col(d^1) = 0$ whereafter Lemma 11 implies $c(y_1^0) = 1$ which is a contradiction. Now assume $c(u^0, u^1) = (1, 0)$. Then, Lemma 12 implies $col(d^0) = 0$ and $col(d^1) = 1$. If the nodes y_i^1, z_i^1 are biased to their unnatural values by type 8 then $c(x^1) = c(\lambda^1)$ due to Lemma 15. Then, at least one of the conditions $w(x^0) \neq c(x^1)$, and $w(x^0) = c(\lambda^1)$ is violated. But this implies that type 8 biases the nodes y_i^1, z_i^1 to their natural value which is a contradiction. Therefore, all nodes y_i^1, z_i^1 are biased to their natural values by type 8. But then Lemma 11 implies that $c(y_1^1) = 1$ which is also a contradiction.

(0, 1): According to Lemma 7 we have $c(y_0^0) = 0$ and $c(z_0^0) = 1$. If $c(u^0, u^1) = (0, 1)$ then Lemma 12 implies $col(d^0) = 1$ and $col(d^1) = 0$ whereafter Lemma 11 implies $c(y_1^0) = 1$ which is a contradiction. Now assume $c(u^0, u^1) = (1, 0)$. Assume for the sake of contradiction that y_0^1 is white. If the nodes y_i^1, z_i^1 are biased to their natural values by type 8 then Observation 15 implies $c(y_1^1) = 1$ which is a contradiction. If they are biased to their unnatural values then $c(z_1^1) = 1$ whereafter $c(y_1^1) = 0$ which is also a contradiction. Thus, y_0^1 is black. Then, Lemma 13 implies $col(d^1) = 0$, but then $c(u^1) = 1$ due to the bias of type 9 which is again a contradiction.

(1, 0): Lemma 7 and Observation 15 together imply $c(y_0^0) = c(z_0^1) = 1$ and $c(z_0^0) = c(y_0^1) = 0$. If $c(u^0, u^1) = (0, 1)$ then Lemma 13 implies $col(d^0) = 0$. But then $c(u^0) = 1$, since u^0 is biased by type 9 to the color of y_0^1 , i. e. to 1, which is a contradiction. Now assume $c(u^0, u^1) = (1, 0)$. Then, $col(d^0) = 0$ and $col(d^1) = 1$ due to Lemma 12. Due to Lemma 16 we have $x^1 = w(x^0)$ in which case all nodes y_i^1, z_i^1 are biased to their natural values by type 8. Then, Lemma 11 implies $c(y_1^1) = 1$ which is a contradiction.

(1, 1): Due to Observation 15 we have $c(y_0^0) = 1$ and $c(z_0^0) = 0$. Since $c(y_1^0) = c(y_1^1) = 1$, Lemma 8 and Lemma 7 together imply that all gates in G_C^0 and G_C^1 are correct and y_i^κ, z_i^κ have their natural values for $\kappa \in \{0, 1\}$, $2 \leq i \leq 2N + 1$. Claim 12 implies $c(\alpha_{j,k}^0) \neq col(I_k(g_j^0))$ and $c(\alpha_{i,j}^1) \neq col(I_j(g_i^1))$ for all i and $1 \leq j \leq 2$. Moreover, we also have $col(g_i^\kappa) \neq col(\hat{g}_i^\kappa)$ for all i, κ . According to type 10, node d^0 is biased to 0 and node d^1 to 1.

In the following, we first show $col(d^0) = 0$ by naming the decisive neighbors of d^0 and showing that their color is black. For this, we distinguish four cases. First, if $c(u^0) = 1$ then u^0 and its counterpart, i. e. the constant 1, are decisive for d^0 . Second, if $c(u^0) = 0$ and $c(z_0^1) = 1$ then z_0^1 and its counterpart, also the constant 1, are decisive since y_0^0 is black and its counterpart is a white constant. Third, if $c(u^0) = 0$, $c(z_0^1) = 0$, and $C(x^0) > C(x^1)$ then the pair of nodes g_i^0, \hat{g}_i^1 with highest index i for $i \leq m$ such that g_i^0 and \hat{g}_i^1 have the same color are both black. Then, Lemma 17 implies $c(u_{i,13}^0) = c(u_{2m+i,15}^1) = 0$ and $c(u_{i,14}^0) = c(u_{2m+i,16}^1) = 1$ – recall that $c(y_{2i-1}^\kappa) = 1$ for $\kappa \in \{0, 1\}$. Moreover, since $g_j^0 \neq g_{2m+j}^1$ for all $j > i$ the same Lemma implies $c(u_{j,13}^0) \neq c(u_{j,14}^0) = g_j^0 \neq g_{2m+j}^1 = c(u_{2m+j,16}^1) \neq c(u_{2m+j,15}^1)$ for all $j > i$. Then, the nodes $u_{i,14}^0$ and $u_{2m+i,16}^1$ are decisive for d^0 . Fourth, if $c(u^0) = 0$, $c(z_0^1) = 0$, and $C(x^0) = C(x^1)$ then the neighbors of type 5 representing the constant 1 adjacent to d^0 via edges of relative weight 1 are decisive for d^0 . In any of the above cases the decisive neighbors of d^0 are black. Since node d^0 is biased to 0 by type 10 Theorem 1 implies $col(d^0) = 0$. Then, the bias of type 9 implies $c(u^0, u^1) = (1, 0)$.

Now we show that $col(d^1) = 1$. For this, we distinguish three cases. First, if $c(y_0^1) = 0$ then y_0^1 and its counterpart, the constant 0, are decisive since $c(u^1) = 0$. Second, if $c(y_0^1) = 1$ and $C(x^0) > C(x^1)$ then the pair of nodes g_i^1, \hat{g}_i^0 with highest index i for $i \leq m$ such that g_i^1 and \hat{g}_i^0 have the same color are both white. Then, Lemma 17 implies $c(u_{i,13}^1) = c(u_{2m+i,15}^0) = 1$, $c(u_{i,14}^1) = c(u_{2m+i,16}^0) = 0$. Moreover, the same Lemma implies $c(u_{j,13}^1) \neq c(u_{j,14}^1) = g_j^1 \neq g_{2m+j}^0 = c(u_{2m+j,16}^0) \neq c(u_{2m+j,15}^0)$ for all $j > i$. Then, the nodes $u_{i,13}^1$ and $u_{2m+i,15}^0$ are decisive for d^1 . Third, $c(y_0^1) = 1$, and $C(x^0) = C(x^1)$ then the neighbors of type 5 representing the constant 0 adjacent to d^1 via edges of relative weight 1 are decisive for d^1 . In any of the above cases the decisive neighbors of d^1 are white. Since node d^1 is biased to 0 by type 10 Theorem 1 implies $col(d^1) = 1$.

According to Lemma 16 we obtain $x^1 = w(x^0)$. Due to our assumption that C only returns its input as better neighbor if the input is locally optimal and since $x^1 = w(x^0)$ the colors of x^0 induce a local optimum of C . This finishes the proof of Theorem 2. \square

5 Smoothed Complexity of Local Max-Cut

We consider the smoothed complexity of local MAX-CUT for graphs with degree $O(\log n)$. Smoothed analysis, as introduced by Spielman and Teng [20], is motivated by the observation that practical data is often subject to some small random noise. Formally, let $\Omega_{n,m}$ be the set of all weighted graphs with n vertices and m edges, in which each graph has maximum degree $O(\log n)$. In this paper, if A is an algorithm on graphs with maximum degree $O(\log n)$, then the smoothed complexity of A with σ -Gaussian perturbation is

$$\text{Smoothed}_A^\sigma(n) = \max_m \max_{G \in \Omega_{n,m}} \mathbb{E}_{\mathbf{x}_m} [T_A(G^{w_{\max} \cdot \mathbf{x}_m})],$$

where $\mathbf{x}_m = (x_1, \dots, x_m)$ is a vector of length m , in which each entry is an independent Gaussian random variable of standard deviation σ and mean 0. $\mathbb{E}_{\mathbf{x}_m}$ indicates that the expectation is taken over vectors \mathbf{x}_m according to the distribution described before, $T_A(G)$ is the running time of A on G , and $G^{w_{\max} \cdot \mathbf{x}_m}$ is the graph obtained from G by adding $w_{\max} \cdot x_i$ to the weight of the i -th edge in G , where w_{\max} is the largest weight in the graph. We assume that the edges are considered according to some arbitrary but fixed ordering.

According to Spielman and Teng, an algorithm A has polynomial smoothed complexity if there exist positive constants c' , n_0 , σ_0 , k_1 , and k_2 such that for all $n > n_0$ and $0 \leq \sigma < \sigma_0$ we have

$$\text{Smoothed}_A^\sigma(n) < c' n^{k_1} \cdot \sigma^{-k_2}.$$

In this paper, we use a relaxation of polynomial smoothed complexity [21], which builds up on Blum and Dungan [3] (see also Beier and Vöcking [4]). According to this relaxation, an algorithm A has probably polynomial smoothed complexity if there exist positive constants c' , n_0 , σ_0 , and α such that for all $n > n_0$ and $0 \leq \sigma < \sigma_0$ we have

$$\max_m \max_{G \in \Omega_{n,m}} \mathbb{E}_{\mathbf{x}_m} [T_A^\alpha(G^{w_{\max} \cdot \mathbf{x}_m})] < \frac{c'n}{\sigma}.$$

Theorem 16. *Let A be some FLIP local search algorithm for local MAX-CUT. Then, A has probably polynomial smoothed complexity on any graph with maximum degree $O(\log n)$.*

Proof. Let $V = \{v_1, \dots, v_n\}$, and denote by d_i the degree of v_i . Furthermore, let $w_{i,j}$ be the weight of edge (v_i, v_j) . Let $m = |E|$, and $\mathbf{x}_m = (x_1, \dots, x_m)$ a vector of Gaussian random variables of standard deviation σ and mean 0. Alternatively, we denote by $x_{i,j}$ the Gaussian random variable which perturbs edge (v_i, v_j) , i. e., $\tilde{w}_{i,j} = w_{i,j} + w_{\max} \cdot x_{i,j}$ represents the weight of (v_i, v_j) in the perturbed graph $G^{w_{\max} \cdot \mathbf{x}_m}$.

In the following, G is an arbitrary graph in $\Omega_{n,m}$ where $m = O(n \log n)$. We show that for any $\delta \in (0, 1)$ there are constants c' , n_0 , σ_0 , k_1 , and k_2 such that for all $n > n_0$ and $0 \leq \sigma < \sigma_0$ we obtain

$$\Pr_{\mathbf{x}_m} [T_A(G^{w_{\max} \cdot \mathbf{x}_m}) < \delta^{-2} c' n^{k_1} \cdot \sigma^{-k_2}] > 1 - \delta. \quad (1)$$

Then, (1) implies the statement of the theorem (cf. [4]).

In order to show the inequality above, we make use of the fact that the sum of k Gaussian random variables with variance σ^2 and mean 0 is a Gaussian random variable with variance $k\sigma^2$ and mean 0. Let X_1, \dots, X_k be k Gaussian random variables with variance σ^2 and mean 0. Furthermore, let a be some real number, and $S \subset \{1, \dots, k\}$. Then, we can state the following claim.

Claim 17. *For some large constant c and any $\delta' \in (0, 1)$*

$$\Pr \left[\left| \sum_{j \in S} X_j - \sum_{j \notin S} X_j - a \right| \leq \frac{\delta' \sigma}{c \cdot 2^k} \right] \leq \delta' \cdot 2^{-k}. \quad (2)$$

Proof. Let $X = \sum_{j \in S} X_j$ and $Y = \sum_{j \notin S} X_j$. Then, X is a Gaussian random variable with variance $|S|\sigma^2$, and Y is a Gaussian random variable with variance $(k - |S|)\sigma^2$. Since X is distributed according to the density function $\frac{1}{\sqrt{2\pi|S|\sigma}} e^{-x^2/(|S|\sigma^2)}$, for any real b it holds that

$$\begin{aligned} \Pr \left[|X - b| \leq \frac{\delta' \sigma}{c \cdot 2^k} \right] &\leq \int_{b - \frac{\delta' \sigma}{c \cdot 2^k}}^{b + \frac{\delta' \sigma}{c \cdot 2^k}} \frac{1}{\sqrt{2\pi|S|\sigma}} e^{-x^2/(|S|\sigma^2)} dx \\ &\leq \frac{1}{\sqrt{2\pi|S|\sigma}} \cdot \frac{2\delta' \sigma}{c \cdot 2^k} \leq \delta' \cdot 2^{-k} \end{aligned}$$

Setting $b = Y + a$ we obtain the claim. \square

In order to show the theorem, we normalize the weights by setting the largest weight to 1, and dividing all other weights by w_{\max} . That is, we obtain some graph G' with weights $w'_{i,j} = w_{i,j}/w_{\max}$. The edge weights of G' are perturbed accordingly by Gaussian random variables with variance σ^2 and mean 0. Clearly, $T_A(G') = T_A(G)$ and $\Pr[T_A(G'^{X_m}) < \delta^{-1}c'n^{k_1} \cdot \sigma^{-k_2}] = \Pr[T_A(G^{w_{\max} \cdot X_m}) < \delta^{-1}c'n^{k_1} \cdot \sigma^{-k_2}]$. Therefore, we consider G' instead of G in the rest of the proof.

In the next step, we show that for an arbitrary but fixed partition P of G' and node v_i , flipping v_i increases (or decreases) the cut by $\Omega(\frac{\delta\sigma}{n2^{d_i}})$, with probability $1 - \delta/2 \cdot n^{-1}2^{-d_i}$. This is easily obtained from Claim (17) in the following way. Define S' to be the set of the neighbors of v_i , which are in the same partition as v_i according to P . Let e_1, \dots, e_{d_i} be the edges incident to v_i , and denote by w_1, \dots, w_{d_i} the weights of these edges in G' . We assume w. l. o. g. that $e_1, \dots, e_{|S'|}$ have both ends in the same partition as v_i , and $S = \{1, \dots, |S'|\}$. Furthermore, let $a = \sum_{j \notin S} w_j - \sum_{j \in S} w_j$, $k = d_i$, and $\delta' = \delta/(2n)$. Applying now Equation (2) we obtain the desired result.

For a node v_i , there are at most $\sum_{i=0}^{d_i} \binom{d_i}{i} = 2^{d_i}$ possibilities to partition the edges into two parts, one subset in the same partition as v_i and the other subset in the other partition. Therefore, by applying the union bound we conclude that any flip of an unhappy v_i increases the cut by $\Omega(\frac{\delta\sigma}{n2^{d_i}})$, with probability at least $1 - \delta/2 \cdot n^{-1}$. Since there are n nodes in total, we may apply the union bound again, and obtain that every flip (carried out by some unhappy node) increases the cut by $\Omega(\frac{\delta\sigma}{n2^{d_i}})$, with probability at least $1 - \delta/2$. Since $d_i = O(\log n)$ and the largest weight in G' is 1, we conclude that the largest cut in G' may have weight $O(n \log n)$. Furthermore, for each i we have $|x_i| \leq l\sqrt{\ln n}$ with probability $1 - O(n^{-l})$ whenever l is large enough (remember that $\sigma < 1$). Let A_1 be the event that there is some x_i with $|x_i| = \omega(\log n)$, and A_2 is the event that there is a node v_i and a partition P such that flipping v_i increases the cut by at most $\tau(\frac{\delta\sigma}{n2^{d_i}})$, where τ is a very small constant. We know that $\Pr[A_1] = n^{-\omega(1)}$ and $\Pr[A_2] < \delta/2$. Thus, as long as $\delta = n^{-O(1)}$, the total number of steps needed by A is at most

$$T_A(G'^{X_m}) = O\left(\frac{n^2 \log^2 n 2^{d_i}}{\delta\sigma}\right) = \frac{n^{O(1)}}{\delta\sigma}$$

with probability $1 - (\Pr[A_1] + \Pr[A_2]) > 1 - \delta$.

Now we consider the case when $\delta = n^{-\Omega(1)}$. Again, let A_1 be the event that there is some x_i with $|x_i| = \omega(\log \delta^{-1})$. Since x_i is a Gaussian random variable, $\Pr[A_1] = \delta^{\omega(1)}$. On the other hand, let A_2 be the event that there is a node v_i and a partition P such that flipping v_i increases the cut by at most $\tau(\frac{\delta\sigma}{n2^{d_i}})$, where τ is a very small constant. Again, $\Pr[A_2] < \delta/2$. Then, the total number of steps needed by A is at most

$$T_A(G'^{X_m}) = O\left(\frac{n^2 \log n \cdot \log \delta^{-1} \cdot 2^{d_i}}{\delta\sigma}\right) = \frac{n^{O(1)} \log \delta^{-1}}{\delta\sigma}$$

with probability $1 - (\Pr[A_1] + \Pr[A_2]) > 1 - \delta$. Note that if $\max_i |x_i| = \log \delta$ then the input size $|G'^{X_m}|$ is $O(n \log \log \delta)$. Hence, the result above does not imply that $T_A(G'^{X_m}) = \frac{|G'^{X_m}|^{O(1)}}{\delta\sigma}$ with probability $1 - \delta$. \square

6 Conclusion and Open Problems

In this paper, we introduced a technique by which we can substitute graphs with certain nodes of unbounded degree, namely so called comparing nodes, by graphs with nodes of maximum degree five such that local optima of the former graphs induce unique local optima of the latter ones. Using this technique, we show that the problem of computing a local optimum of the MAX-CUT problem is \mathcal{PLS} -complete even on graphs with maximum degree five. We do not show that our \mathcal{PLS} -reduction is tight, but the tightness of our reduction would not result in the typical knowledge gain anyway since the properties that come along with the tightness of \mathcal{PLS} -reductions, namely the \mathcal{PSPACE} -completeness of the standard algorithm problem and the existence of instances that are exponentially many improving steps away from any local optimum, are already known for the maximum degree four [15]. The obvious remaining question is to ask for the complexity of local MAX-CUT on graphs with maximum degree four. Is it in \mathcal{P} ? Is it \mathcal{PLS} -complete? Another important question is whether local MAX-CUT has in general probably polynomial smoothed complexity. Unfortunately, the

methods used so far seem not to be applicable to show that in graphs with super-logarithmic degree the local MAX-CUT problem has probably polynomial smoothed complexity (cf. also [18]).

Acknowledgement. We thank Dominic Dumrauf, Martin Gairing, Martina Hüßmann, Burkhard Monien, and Rahul Savani for helpful suggestions.

References

1. David Arthur, Bodo Manthey, and Heiko Röglin: *k-Means has polynomial smoothed complexity*. FOCS'09, 405-414, 2009.
2. Heiner Ackermann, Heiko Röglin, Berthold Vöcking: *On the impact of combinatorial structure on congestion games*. Journal of the ACM (JACM), Vol. 55(6), Art. 25, 2008.
3. Avrim Blum and John Dunagan: *Smoothed analysis of the perceptron algorithm for linear programming*. SODA, 905-914, 2002.
4. René Beier and Berthold Vöcking: *Typical properties of winners and losers in discrete optimization*. STOC, 343-352, 2004.
5. Matthias Englert, Heiko Röglin, Berthold Vöcking: *Worst case and probabilistic analysis of the 2-Opt algorithm for the TSP*. SODA, 1295-1304, 2006.
6. Alex Fabrikant, Christos H. Papadimitriou, Kunal Talwar: *The complexity of pure Nash Equilibria*. STOC, 604-612, 2004.
7. Martin Gairing, Rahul Savani: *Computing stable outcomes in hedonic games*. In: Kontogiannis, S., Koutsoupias, E., Spirakis, P.G. (eds.), SAGT, LNCS 6386, 174-185, 2010.
8. Michael R. Garey, David S. Johnson: *Computers and intractability, a guide to the theory of \mathcal{NP} -completeness*. Freeman, New York, NY, 1979.
9. David S. Johnson, Christos H. Papadimitriou, Mihalis Yannakakis: *How easy is local search?* Journal of Computer and System Sciences 37 (1), 79-100, 1988.
10. Mark W. Krentel. *Structure in locally optimal solutions*. FOCS, 216-221. 1989.
11. Jonathan A. Kelner and Evdokia Nikolova: *On the hardness and smoothed complexity of quasi-concave minimization*. FOCS, 472-482, 2007.
12. Richard E. Ladner: *The circuit value problem is log space complete for P*. SIGACT News, 7:1, 18-20, 1975.
13. Martin Loebl: *Efficient maximal cubic graph cuts*. In: Leach Albert, J., Monien, B., Rodríguez-Artalejo, M. (eds.) ICALP, LNCS 510, 351-362, 1991.
14. Burkhard Monien, Dominic Dumrauf, Tobias Tscheuschner: *Local search: Simple, successful, but sometimes sluggish*. In: Abramsky, S., Gavioille, C., Kirchner, C., Meyer auf der Heide, F., Spirakis, P.G. (eds.) ICALP, LNCS 6198, 1-17, 2010.
15. Burkhard Monien, Tobias Tscheuschner: *On the power of nodes of degree four in the local max-cut problem*. In: Calamoneri, T., Diaz, J. (eds.) CIAC, LNCS 6078, 264-275, 2010.
16. Svatopluk Poljak: *Integer linear programs and local search for max-cut*. SIAM Journal on Computing 21(3), 450-465, 1995.
17. Svatopluk Poljak, Zsolt Tuza: *Maximum cuts and largest bipartite subgraphs*. Combinatorial Optimization, American Mathematical Society, Providence, RI, 181-244, 1995.
18. Heiko Röglin: *personal communication*, 2010.
19. Heiko Röglin and Berthold Vöcking: *Smoothed analysis of integer programming*. Math. Program., 110 (1), 21-56, 2007.
20. Daniel Spielmann, Shang-Hua Teng: *Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time*. Journal of the ACM (JACM), 51(3), 385-463, 2004.
21. Daniel Spielmann, Shang-Hua Teng: *Smoothed analysis: an attempt to explain the behavior of algorithms in practice*, Commun. ACM 52 (10), 76-84, 2009.
22. Alejandro A. Schäffer, Mihalis Yannakakis: *Simple local search problems that are hard to solve*. SIAM Journal on Computing 20(1), 56-87, 1991.
23. Roman Vershynin: *Beyond Hirsch Conjecture: Walks on Random Polytopes and Smoothed Complexity of the Simplex Method*. FOCS, 133-142, 2006.

A Appendix

A.1 Definition of the nodes of type I and III and proof of Lemma 1

We introduce the types I and III of the nodes of degree ≤ 4 , since their use simplifies the proof.

Definition 4. For a node u and edges a_u, b_u, c_u, d_u incident to u with $w(a_u) \geq w(b_u) \geq w(c_u) \geq w(d_u)$ we distinguish the following types for u :

- type I: if $w(a_u) > w(b_u) + w(c_u) + w(d_u)$
- type III: if $w(a_u) + w(d_u) < w(b_u) + w(c_u)$

Observation 18. For a graph $G = (V, E)$, a partition P , a node u , and edges a_u, b_u, c_u, d_u incident to u with $w(a_u) \geq w(b_u) \geq w(c_u) \geq w(d_u)$ the following two conditions hold:

- If u is of type I then u is happy in G_P if and only if a_u is in the cut.
- If u is of type III then u is happy in G_P if and only if at least two of the edges a_u, b_u, c_u are in the cut.

The original statement of Lemma 1 in [15] contained the following two properties.

Theorem 19. (a) Finding a local optimum of MAX-CUT on graphs containing only nodes of type I and III is P-hard with respect to logspace-reduction.
 (b) Let $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ be a function and C be a boolean circuit with N gates which computes f . Then, using $O(\log N)$ space, one can compute a graph $G^C = (V^C, E^C)$ containing only nodes of type I and III among which there are nodes $s_1, \dots, s_n, t_1, \dots, t_m \in V^C$ of degree one such that $f(c_P(s)) = c_P(t)$ in every local optimum P of G^C .

Proof. At first, we show property (a). We reduce from the P-complete problem circuit-value [12]. An instance of the circuit-value problem is a Boolean circuit C together with an assignment for the inputs of C . The problem asks for the output of C on the given input assignment. For the gates, we assume w. l. o. g. that they are either NOR-gates with a fanin of two and fanout of one or NOT-gates with a fanin of one and fanout of at most two. We order the gates g_N, g_{N-1}, \dots, g_1 topologically such that if g_i is an input of g_j then $i > j$. We let g_m, \dots, g_1 be the output of the C and assume w. l. o. g. that the gates g_N, \dots, g_{N-n+1} are NOT-gates, that they are the only gates in which the inputs of the circuit C occur, and that the output gates are NOT-gates with fanout zero. Let $I_1(g_i)$ and $I_2(g_i)$ be the gates which are the inputs of a NOR-gate g_i for $1 \leq i \leq N - n$, $I(g_i)$ be the input of a NOT-gate g_i for $1 \leq i \leq N - n$, and $value(g_i)$ be the corresponding value of the assignment of the input of g_i with $N - n + 1 \leq i \leq N$.

We construct a graph $T = (V, E)$ with weights $w : E \rightarrow \mathbb{N}$ from C in the following way. The set of nodes is $V = \{v_1, \dots, v_{3N+1}\}$ and the set E as follows:

- (i) For each NOR-gate g_i with $1 \leq i \leq N - n$ we have the following edges of weight 2^i : $\{v_i, v_j\} \in E$ if $I_k(g_i) = g_j$ for $1 \leq k \leq 2, 1 \leq j \leq N$, and $\{v_i, v_{N+2i}\} \in E$.
- (ii) For each NOT-gate g_i with $1 \leq i \leq N - n$ with $I(g_i) = g_j$ for $1 \leq j \leq N$ we have $\{v_i, v_j\} \in E$ with $w(\{v_i, v_j\}) = 2^i$.
- (iii) For each g_i with $N - n + 1 \leq i \leq N$ we have the following edge of weight 2^i : if $value(g_i) = 1$ then $\{v_i, v_{N+2i}\} \in E$, otherwise $\{v_i, v_{N+2i-1}\} \in E$.
- (iv) $\{v_i, v_{i+1}\} \in E$ for all $N + 1 \leq i \leq 3N$ with a weight of 2^i .

Then, each node v_i for $N + 1 \leq i \leq 3N$ has a degree of at most three, is of type I, and the heaviest edge incident to v_i is $\{v_i, v_{i+1}\}$. The node v_{3N+1} has a degree of one, is also of type I, and its heaviest edge is $\{v_{3N+1}, v_{3N}\}$. Each node v_i for $N - n + 1 \leq i \leq N$ has a degree of two, is of type I, and its heaviest edge is the one described in (iii). Each node v_i for $1 \leq i \leq N - n$, for which g_i is a NOT-gate, has a degree of at most three, is of type I, and its heaviest edge is the edge of weight 2^i described in (ii). However, if g_i is a NOR-gate then v_i has a degree of four, is of type III, and the edges with influence on it are the three edges of weight 2^i described in (i).

Consider a local optimum P of T . Due to the symmetry of the local MAX-CUT problem we may assume w. l. o. g. that $c_P(v_{3N}) = 1$. In the following, we use Observation 18 to derive the colors of the remaining

nodes as induced by their types. First, $c_P(v_i) \neq c_P(v_{i+1})$ for all $N+1 \leq i \leq 3N$. Thus, $c_P(v_{N+2i}) = 1$ and $c_P(v_{N+2i-1}) = 0$ for all $1 \leq i \leq N$ and $c_P(v_{3N+1}) = 0$. Then, for each $N-n+1 \leq i \leq N$ we have $c_P(v_i) = 1$ if $\text{value}(g_i) = 1$ and $c_P(v_i) = 0$ otherwise, i. e. the colors of the nodes v_i for $N-n+1 \leq i \leq N$ correspond to the assignment for the inputs of C . Now consider the nodes v_i for $1 \leq i \leq N-n$. If g_i is a NOT-gate with $I(g_i) = g_j$ for $m+1 \leq j \leq N$ then $c_P(v_i) \neq c_P(v_j)$, i. e. the color of v_i corresponds to the output of a NOT-gate w. r. t. the color of v_j . Finally, if g_i is a NOR-gate with $I_1(g_i) = g_k$ and $I_2(g_i) = g_j$ for $m+1 \leq j < k \leq N$ then $c_P(v_i) = 1$ if and only $c_P(v_j) = c_P(v_k) = 0$ since v_i is of type III and its neighbor v_{N+2i} is already known to be black. Thus, the color of v_i corresponds to the output of a NOR-gate with respect of the colors of v_j and v_k . Therefore, the color of each node v_i for $1 \leq i \leq N-n$ corresponds to the output of g_i in C . In particular, the colors of v_1, \dots, v_m correspond to the output of C .

In the following we show that our reduction is in logspace. The number of edges in T is linear in N since each node has maximum degree four. The weights of the edges are powers of two. Thus, we only need to store the exponents of the weights. If we write an edge weight to the output tape then we first write the “1” for the most significant bit and then we write as often a “0” as determined by the exponent.

Now we show (b). We let $G^C = (V^C, E^C)$ be the graph obtained from T by omitting the edges described in (iii). Furthermore, we let $s_i = v_{N-n+i}$ for $1 \leq i \leq n$ and $t_j = v_j$ for $1 \leq j \leq m$. Then, the nodes s_i and t_j are of degree one. As in the proof for (a) we get $f(c_P(s)) = c_P(t)$. \square